

Школа: Инженерная школа информационных технологий и робототехники
 Направление подготовки: 09.03.01 Информатика и вычислительная техника
 Отделение школы (НОЦ): Отделение информационных технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка мобильного образовательного приложения под ОС Android

УДК 004.455:004.85

Студент

Группа	ФИО	Подпись	Дата
8В7Б	Андросова Анна Сергеевна		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Саврасов Федор Витальевич	К.Т.Н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСНГ	Дукарт Сергей Александрович	К.И.Н.		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ООД	Мезенцева Ирина Леонидовна	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Погребной Александр Владимирович	К.Т.Н.		

**Планируемые результаты обучения по основной образовательной
программе подготовки бакалавров 09.03.01 «Информатика и
вычислительная техника»**

Код результатов	Результат обучения (выпускник должен быть готов)
P1	Применять глубокие естественнонаучные, математические и инженерные знания для создания и обработки новых материалов
P2	Применять глубокие знания в области современных технологий машиностроительного производства для решения междисциплинарных инженерных задач
P3	Ставить и решать инновационные задачи инженерного анализа, связанные с созданием и обработкой материалов и изделий, с использованием системного анализа и моделирования объектов и процессов машиностроения
P4	Разрабатывать технологические процессы, проектировать и использовать новое оборудование и инструменты для обработки материалов и изделий, конкурентоспособных на мировом рынке машиностроительного производства
P5	Проводить теоретические и экспериментальные исследования в области современных технологий обработки материалов, нанотехнологий, создания новых материалов в сложных и неопределенных условиях
P6	Внедрять, эксплуатировать и обслуживать современные высокотехнологичные линии автоматизированного производства, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья и безопасности труда на машиностроительном производстве, выполнять требования по защите окружающей среды
P7	Использовать глубокие знания по проектному менеджменту для ведения инновационной инженерной деятельности с учетом юридических аспектов защиты интеллектуальной собственности
P8	Активно владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности
P9	Эффективно работать индивидуально, в качестве члена и руководителя группы, состоящей из специалистов различных направлений и квалификации, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации
P10	Демонстрировать глубокие знания социальных, этических и культурных аспектов инновационной инженерной деятельности, компетентность в вопросах устойчивого развития
P11	Самостоятельно учиться и непрерывно повышать квалификацию в течение всего периода профессиональной деятельности

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа: Инженерная школа информационных технологий и роботехники
 Направление подготовки: 09.03.01 Информатика и вычислительная техника
 Отделение школы (НОЦ): Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП
 _____ Погребной А.В.
 (Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы
(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8В7Б	Андросовой Анне Сергеевне

Тема работы:

Разработка мобильного образовательного приложения под ОС Android	
Утверждена приказом директора (дата, номер)	№ 32-1/с от 01.02.2021

Срок сдачи студентом выполненной работы:	
--	--

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<p>Исходные данные к работе</p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Техническое задание к реализации мобильного образовательного приложения «IT Way» под операционную систему Android для отображения разнородного контента, такого как видео, новости, подкасты.</p>
<p>Перечень подлежащих исследованию, проектированию и разработке вопросов</p> <p><i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> 1. Аналитический обзор 2. Теоретический анализ 3. Проектирование мобильного приложения 4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение 5. Социальная ответственность

Перечень графического материала (с точным указанием обязательных чертежей)	Презентация в формате *.pptx
--	------------------------------

Консультанты по разделам выпускной квалификационной работы
(с указанием разделов)


Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Дукарт Сергей Александрович
Социальная ответственность	Мезенцева Ирина Леонидовна

Названия разделов, которые должны быть написаны на русском и иностранном языках:

1. Обзор литературы
2. Объект и методы исследования
3. Теоретический анализ
4. Результаты проведенной разработки
5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение
6. Социальная ответственность

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	
---	--

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Саврасов Федор Витальевич	К.Т.Н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8В7Б	Андросова Анна Сергеевна		

Министерство науки и высшего образования Российской Федерации
 федеральное государственное автономное
 образовательное учреждение высшего образования
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа: Инженерная школа информационных технологий и роботехники
 Направление подготовки: 09.03.01 Информатика и вычислительная техника
 Уровень образования: бакалавриат
 Отделение школы (НОЦ): Отделение информационных технологий
 Период выполнения: весенний семестр 2020/2021 учебного года

Форма представления работы:

Бакалаврская работа

(бакалаврская работы, дипломный проект/работа, магистерская диссертация)


КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	
--	--

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	Основная часть	80
	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
	Социальная ответственность	10

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Саврасов Федор Витальевич	к.т.н.		

СОГЛАСОВАНО:

Руководитель ООП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Погребной Александр Владимирович	к.т.н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8В7Б	Андросовой Анне Сергеевне

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.01 Информатика и вычислительная техника

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Бюджет проекта – не более 224 465 руб., в т.ч. затраты по оплате труда – не более 151 503 руб.
2. <i>Нормы и нормативы расходования ресурсов</i>	Значение показателя интегральной ресурсоэффективности – не мене 5 баллов из 5
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Коэффициент отчислений на уплату во внебюджетные фонды 30 %

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	Описание потенциальных потребителей; SWOT-анализ.
2. <i>Планирование и формирование бюджета научных исследований</i>	Формирование плана и графика разработки: – определение структуры работ; – определение трудоемкости работ; – создание диаграммы Ганта. Формирование бюджета затрат на разработку: – расчет затрат на специальное оборудование; – расчет заработной платы (основной и дополнительной); – расчет страховых взносов; – расчет накладных расходов.
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	Определение потенциального эффекта разработки.

Перечень графического материала (с точным указанием обязательных чертежей):

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. Альтернативы проведения НИ
4. График проведения и бюджет НИ

5. Оценка ресурсной, финансовой и экономической эффективности НИ

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП ТПУ	Дукарт Сергей Александрович	к.и.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8В7Б	Андросова Анна Сергеевна		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8В7Б	Андросовой Анны Сергеевны

Школа	Инженерная школа информационных систем и робототехники	Отделение (НОЦ)	Отделение информационных технологий
Уровень образования	Бакалавриат	Направление/специальность	09.03.01 «Информатика и вычислительная техника»

Тема ВКР:

Разработка мобильного образовательного приложения под ОС Android	
Исходные данные к разделу «Социальная ответственность»:	
1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения	Работа посвящена разработке мобильного приложения под ОС Android для скилл-шеринг сообщества «IT Way». Приложение позволит объединить весь разнородный контент (статьи, новости, видео и подкасты), производимый сообществом, в одном месте для комфортного получения образовательной информации пользователем.
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности: <ul style="list-style-type: none"> – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны. 	<ul style="list-style-type: none"> – Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 27.12.2018); – ГОСТ 12.2.032-78 Рабочее место при выполнении работ сидя; – ГОСТ 21889-76 Система "Человек-машина". Кресло человека-оператора; – ГОСТ 22269-76. Система «человек-машина». Рабочее место оператора. Взаимное расположение элементов рабочего места; – ГОСТ Р 50923-96. Дисплеи. Рабочее место оператора.

2. Производственная безопасность: 2.1. Анализ выявленных вредных и опасных факторов 2.2. Обоснование мероприятий по снижению воздействия	Вредные и опасные факторы: – Отсутствие или недостаток необходимого естественного освещения; – Умственное перенапряжение; – Монотонность труда; – Физическая перегрузка, связанная с работой сидя; – Опасные факторы, связанные с электрическим током (например, повышенные значения напряжения в сети).
3. Экологическая безопасность:	– Выброс углекислого газа при производстве тепло- и электроэнергии; – Воздействие на литосферу в результате образования бытового мусора и отходов при поломке техники;
4. Безопасность в чрезвычайных ситуациях:	– Возможные ЧС: угроза пандемии, компьютерный терроризм; – Наиболее типичная ЧС – пожар.

Дата выдачи задания для раздела по линейному графику	
--	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Мезенцева Ирина Леонидовна	-		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8В7Б	Андросова Анна Сергеевна		

Реферат

Пояснительная записка содержит 105 страниц, 27 рисунков, 14 таблиц, 25 источников и 1 приложение.

Цель работы – разработка мобильного образовательного приложения для сообщества «IT Way».

Ключевые слова: операционная система, Flutter SDK, виджет, API, Firebase, контент, пакет.

Объект исследования данного проекта – мобильное приложение.

В процессе выполнения работы использовались среда разработки Visual Studio Code, программа реализована на языке Dart с использованием Flutter SDK.

В результате выполнения работы было разработано мобильное приложение под ОС Android, которое позволяет объединить разнородный контент в одном месте, для комфортного потребления пользователем.

В первом разделе представлен анализ предметной области, рассмотрен рынок мобильных технологий.

Во втором разделе представлено техническое задание для данного приложения.

Третий раздел представляет собой технический анализ, направленный на выбор среды разработки, языка программирования и фреймворка, хостинга и библиотек.

Четвертый раздел описывает результаты разработки данного приложения.

Пятый раздел представляет собой анализ в области проектного и финансового менеджмента.

В шестом разделе рассмотрены аспекты социальной ответственности.

Область применения: участники сообщества «IT Way» и люди, а именно школьники и студенты, кто только начинает свой путь в IT-сферу.

Определения, обозначения, сокращения и нормативные ссылки

ОС (OS) – операционная система;

ПЭВМ – Персональная электронно-вычислительная машина;

API – Application Programming Interface – программный интерфейс приложения; описание способов, которыми одна программа может взаимодействовать с другой программой;

Firebase – облачная СУБД класса NoSQL, позволяющая разработчикам приложений хранить и синхронизировать данные между несколькими клиентами;

Виджет – визуальный компонент в Flutter;

Flutter – открытый набор средств разработки мобильного пользовательского интерфейса, для разработки с Flutter используется язык программирования под названием Dart;

Контент – информационное содержание (тексты, графическая, звуковая информация и др.);

Пакет – это разделяемые библиотеки или модули.

Содержание

Введение	15
1 Обзор литературы.....	16
2 Объект и методы исследования	17
2.1 Техническое задание	17
2.1.1 Общее положение.....	17
2.1.2 Требования к системе	17
2.1.2.1 Общие требования	17
2.1.2.2 Требования к программе.....	18
2.1.3 Пользовательский интерфейс и его описание.....	18
3 Теоретический анализ	20
3.1 Язык программирования и фреймворк.....	20
3.2 Android Studio и Visual Studio Code	22
3.2.1 Функциональность	23
3.2.2 Поиск.....	24
3.2.3 Контроль версий	25
3.3 Firebase.....	25
3.4 Пакеты Dart и Flutter.....	27
3.4.1 Firebase_auth, provider, firebase_core.....	27
3.4.2 Bloc	29
3.4.3 Audioplayers, audio_service	30
3.4.4 Youtube_player_flutter.....	31
4 Результаты проведенной разработки.....	32
5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	43

5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	43
5.1.1 Потенциальные потребители результатов исследования.....	43
5.1.2 Анализ конкурентных технических решений.....	43
5.1.3 Анализ по технологии QuaD	45
5.1.4 SWOT-анализ.....	46
5.2 Определение возможных альтернатив проведения научных исследований	47
5.3 Планирование научно-исследовательских работ	47
5.3.1 Структура работ в рамках научного исследования	47
5.3.2 Определение трудоемкости выполнения работ	48
5.3.3 Разработка графика проведения научного исследования	49
5.3.4 Бюджет научно-технического исследования (НТИ).....	53
5.3.4.1 Расчет материальных затрат НТИ	53
5.3.4.2 Расчет затрат на специальное оборудование для научных работ	53
5.3.4.3 Основная заработная плата исполнителей темы.....	54
5.3.4.4 Дополнительная заработная плата исполнительской системы	56
5.3.4.5 Отчисления во внебюджетные фонды (страховые отчисления)	57
5.3.4.6 Накладные расходы.....	57
5.3.4.7 Формирование бюджета затрат научно-исследовательского проекта	58
5.4 Определение ресурсной, финансовой, бюджетной, социальной и экономической эффективности исследования.....	58
Вывод по разделу	60
6 Социальная ответственность.....	62

6.1 Правовые и организационные вопросы обеспечения безопасности	62
6.1.1 Специальные правовые нормы трудового законодательства ..	62
6.1.2 Основные требования к компоновке рабочей зоны.....	63
6.2 Производственная безопасность.....	65
6.2.1 Анализ опасных и вредных производственных факторов	65
6.2.1.1Отсутствие или недостаток необходимого естественного освещения	65
6.2.1.2Умственное перенапряжение	66
6.2.1.3Монотонность труда	67
6.2.1.4Физическая перегрузка, связанная с работой сидя	67
6.2.1.5Опасные факторы, связанные с электрическим током.....	67
6.2.2 Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов на работающего	68
6.3 Экологическая безопасность.....	68
6.4 Безопасность в чрезвычайных ситуациях.....	69
Вывод по разделу	70
Заключение	72
Список источников	73
Приложение А	77

Введение

В современной России все чаще появляются и становятся популярными сообщества единомышленников. Суть сообществ (или комьюнити) состоит в следующем: это группа людей, которая совместно, добровольно, и в отсутствии материального стимулирования, занимается деятельностью, в процессе которой могут достигаться общие для участников группы и цели.

Одним из таких примеров является скилл-шеринг сообщество «IT Way». Команда «IT Way», состоящая из программистов, менеджеров, тестировщиков, также специалистов других отраслей: педагогов, журналистов, фотографов, художников, студентов и школьников, делает образовательные конференции, мероприятия и подкасты для людей, в основном школьников и студентов, кто только встал на путь IT и хочет разобраться, в каком направлении ему двигаться в данной сфере.

Производимым контентом сообщества являются: видео, новости, прямые трансляции и подкасты. К сожалению, размещение всего контента на данный момент невозможно. YouTube подходит только для видео и прямых трансляций, телеграмм только для общения и обсуждения, ВКонтакте только для размещения постов, объявлений. В итоге, потребителю требуется какое-то одно место, где он сможет получить контент с удобством.

Таким образом, объектом исследования выпускной квалификационной работы является разработка мобильного приложения на базе операционной системы Android с возможностью комфортного потребления информации пользователями. Приложение должно воспроизводить видео и подкасты, а также содержать новости.

1 Обзор литературы

Согласно данным социального отчета об обзоре и аналитических сведений об использовании социальных сетей, электронной коммерции, Интернета и мобильных устройств по всему миру «Цифровые технологии 2021 года: последние сведения о цифровом состоянии» команд We are social и Hootsuite, мобильным телефоном на сегодняшний день пользуется примерно 67% всего населения Земли. При этом, данные ведущей компании в области глобального маркетинга GWI показывают, что 9 из 10 пользователей предпочитают мобильный Интернет, но также 60% используют ноутбук или настольный компьютер для входа в Интернет [1]. Также стоит обратить внимание на отчет исследовательской компании Canalys, в котором сказано, что мировые поставки смартфонов выросли на 27% в сравнении с прошлым годом, и доля рынка, занимаемая Android OS в России, составляет 75,35%. [2]

Следовательно, можно сказать, что мобильные технологии зарекомендовали себя на мировом рынке и поддаются тенденции роста, соответственно, мобильные приложения пользуются спросом.

Мобильное приложение – это ПО для мобильных устройств, написанное под конкретную ОС. Мобильные устройства удобны в том, что обеспечивают быстроту доступа к контенту, оплате, общению, навигации, банковским услугам, и множеству других сервисов.

2 Объект и методы исследования

В данном разделе приводятся результаты ранее выполненных учебных исследовательских разработок в рамках обучения, напрямую связанные с данной работой.

2.1 Техническое задание

2.1.1 Общее положение

Полное наименование системы: мобильное приложение «IT Way».

Термины и определения:

Новость – содержит текстовую новостную информацию и изображение(-ия).

Видео – содержит видео и прямые трансляции.

Подкасты – содержит подкасты.

Заметки – содержит контент, сохраненный пользователем (понравившийся ему).

Профиль – регистрация или вход в систему, переход в телеграмм и на сайт.

Весь контент представлен в виде карточек, по нажатию на которые потребитель переходит к самой информации, и доступ к которым имеется только при подключении к Интернету. Некоторый тип контента можно комментировать.

2.1.2 Требования к системе

2.1.2.1 Общие требования

Перечень функций:

- возможность добавлять контент в закладки;
- возможность комментировать, общаться в комментариях;

- просмотр видео;
- прослушивание подкаста в фоновом режиме;
- возможность изменять скорость воспроизведения;
- регистрация;
- возможность делиться со сторонними приложениями.

2.1.2.2 Требования к программе

Для надлежащей работоспособности программы, телефону, на котором производится запуск мобильного приложения, достаточно наличия Android версии 5.0 и выше и интернета. Визуальное представление – двумерное.

Мобильное приложение должно быть адаптировано под следующие рабочие станции – мобильное устройство (широкоформатный экран – для видео, стандартный – для всего остального).

Для разработки мобильного приложения используются язык программирования Dart, Flutter SDK, Firebase, и различные пакеты. Среда разработки: Visual Code.

2.1.3 Пользовательский интерфейс и его описание

Стартовая страница при входе в приложение представляет собой «Подкасты», предоставляет переход к разным типам контента и профилю. По нажатию на любую карточку, пользователь оказывается на странице с ее содержанием (также это доступно с других страниц, переход к которым расположен в нижней части экрана). На странице «Подкасты» по нажатию на карточку осуществляется переход к прослушиванию (подкаст сразу начинает проигрываться). Открываются функции фонового прослушивания и изменения скорости воспроизведения. Комментарии не предусмотрены.

На странице «Видео» по нажатию на карточку осуществляется переход к проигрыванию видео в плеере YouTube (видео сразу начинает проигрываться).

Предусмотрены комментарии, дополнительно представлена подборка «Смотрите также». Страница «Новости» аналогична.

Когда пользователь заходит в приложение, авторизация не требуется. Она нужна для того, чтобы оставлять комментарии, отвечать на комментарии других пользователей, и поставить оценки комментариям. Чтобы зарегистрироваться, необходимо перейти на страницу пользователя (переход осуществляется по нажатию на иконку аватара). Также на этой странице доступны переходы на сайт сообщества и к чату в телеграмме.

3 Теоретический анализ

3.1 Язык программирования и фреймворк

Как было сказано ранее, мир мобильных технологий быстро развивается, из-за чего почти каждой компании необходимо свое мобильное приложение, чтобы оставаться конкурентоспособными на рынке. Более того: как правило, компания ищет возможность создавать мобильное приложение под две популярные системы (Android и IOS) с меньшими ресурсами и более высокой скоростью разработки.

Apple и Google предоставляют собственные инструменты и технологии для создания приложений. Так, разработчики приложений для IOS могут создавать приложения, используя Xcode и Swift, в то время как разработчики Android использует Android Studio, а также Kotlin или Java. Однако это требует от программиста освоения двух совершенно разных технологий. В результате, в настоящее время начинает внедряться кроссплатформенное решение вместо нативного, чтобы быстрее создавать приложение для IOS и Android, используя только один язык программирования.

Одним из удачных кроссплатформенных решений является фреймворк React Native. Есть еще одна более новая технология под названием Flutter SDK. Она была представлена в 2017 году и официально выпущена только в декабре 2018 года, позволяет разработчикам писать кроссплатформенное решение, которое можно запускать на различных системах с общей базой кода на Dart, который компилируется в бинарный код, за счет чего достигается скорость выполнения операций, сравнимая с нативными решениями.

Flutter – это набор инструментов разработки, содержащий все библиотеки и инструменты, а также документацию, то есть всё необходимое для создания программного обеспечения. Во Flutter используется язык программирования Dart. Google позиционирует его как альтернативу JavaScript со строгой типизацией, высокой производительностью и гибкостью. Основная идея Flutter вращается вокруг виджетов – весь пользовательский интерфейс состоит из

объединения различных виджетов, каждый из которых определяет структурный элемент, например, кнопку или меню стилистических элементов (шрифт или цветовую схему, аспект макета, отступы, и т.д.). Приложения строятся из них как из конструкторного набора.

Далее разберем некоторые аспекты технического плана, сравнивая Flutter и React Native. В таблице 1 отражены главные критерии оценки [3].

Таблица 1 – сравнение Flutter и React Native.

Критерии оценки	Описание	Flutter	React Native
Сборка и развертывание	Обе платформы используют инструменты мобильной операционной системы для сборки: Gradle и Xcode build для Android и IOS соответственно. Разница отсутствует. Отличием является скорость сборки. Flutter опережает React Native по сборке приложения «Hello Word» примерно в 3 раза.	+	-
Доступность библиотек	Так как технологии появились в разное время, и React Native является старшей, то список качественных библиотек значительно больше.	-	+
Виджеты	На данный момент Flutter немного уступает в каких-то компонентах, но на данную работу это не влияет.	+	+
Инструменты и тестирование	Технологии не уступают друг другу. При этом у Flutter есть одна особенность – это быстрая перезагрузка с сохранением состояния. Вы можете внести изменения в приложении во время его работы, и она перезагрузит измененный код приложения, позволив продолжить его с того места, на котором оно было остановлено (менее секунды).	+	-

Производительность	<p>React Native использует родные виджеты платформы (Native Views) и передает события через JavaScript. Это влияет на производительность уровня представления.</p> <p>Flutter, с другой стороны, рендерит все, используя собственный 2D-движок Skia, избегая какого-либо специального соединения между представлениями и другим кодом. Это делает рендеринг очень быстрым.</p> <p>Также React Native уступает по времени запуска.</p>	++	+-
Опыт разработчика	Здесь говорится об опыте студента, что является важным для выбора технологии.	+	-
Возможности сети	Обе технологии поддерживают парсинг JSON, необходимый для данной работы.	+	+
Возможности медиа	Предоставляются установочными пакетами.	+	+
Безопасность	С Flutter гораздо сложнее изменить приложение, так как Dart-код заранее компилируется в бинарный код для целевой архитектуры.	+	-

Проанализировав данную таблицу, можно сделать вывод, что для данного проекта больше подходит фреймворк Flutter.

3.2 Android Studio и Visual Studio Code

Когда дело доходит до выбора инструментов, которые мы используем для разработки приложений на Flutter, обычно это сводится к двум основным IDE (интерактивная среда разработки) – к Visual Studio Code или к Android Studio.

Visual Studio Code – это один из наиболее популярных редакторов кода, разработанный корпорацией Microsoft [4]. Android Studio — интегрированная среда разработки производства Google [5].

Сравнение представит субъективную суммарную оценку по трем параметрам, а именно:

- 1 функциональность – насколько легко управляться в среде, переключая вкладки, работать над кодом;
- 2 поиск файлов или элементов кода;
- 3 контроль версий – насколько просто видеть изменения.

Попробовав обе среды разработки, было принято решение использовать Visual Studio Code.

3.2.1 Функциональность

Visual Studio Code – это редактор, который предпочитает простоту бесконечному набору возможностей (рисунок 1). Он выглядит интуитивно понятнее, чем Android Studio (рисунок 2). Также предоставляется возможность открыть два файла, разделив их посередине. Практический эффект этого заключается в том, что если разработчик осуществляет совместное проектирование между своих сервисов и уровней пользовательского интерфейса, ему не придется постоянно переключаться между вкладками.

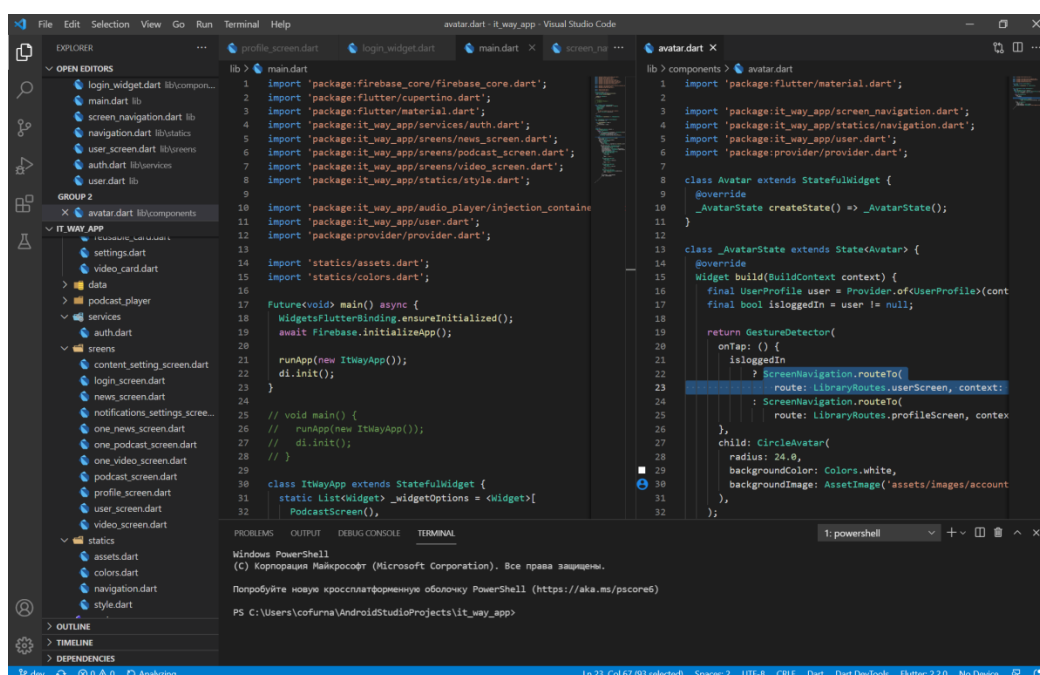


Рисунок 1 – Visual Studio Code.

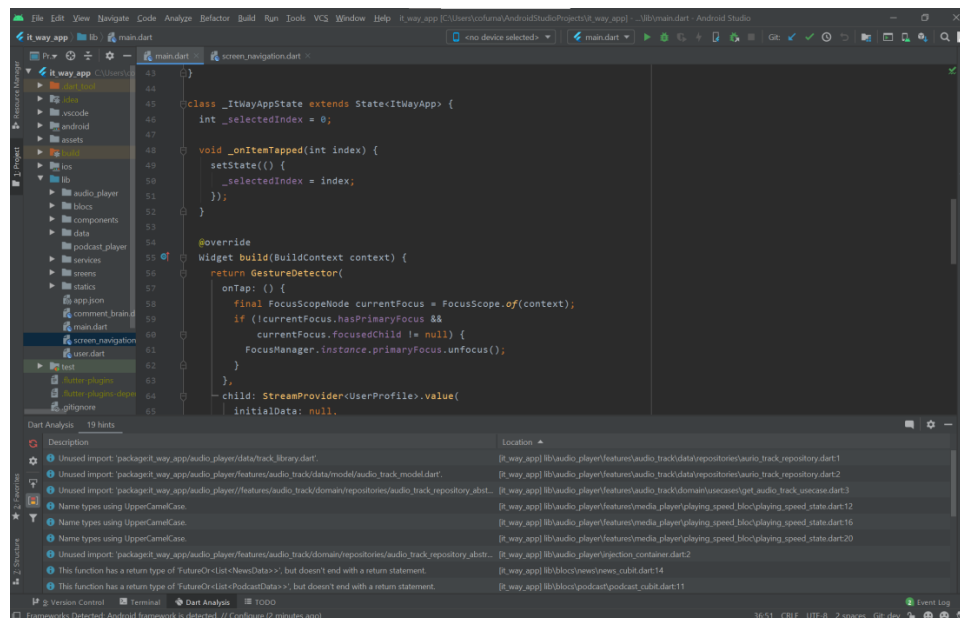


Рисунок 2 – Android Studio.

3.2.2 Поиск

Visual Studio Code имеет возможность осуществлять поиск по всему проекту, когда пользователь нажимает **Ctrl+Shift+F**, аналогичная операция поддерживается вторым IDE. Как можно увидеть, в левой части окна, представленного на рисунке 3, отображаются все результаты. Программист может перемещаться по ним вперед и назад, что является достаточно функциональным. Также поиск может осуществляться в каждом отдельном файле. Обе функции поиска не отрывают разработчика от анализа проекта.

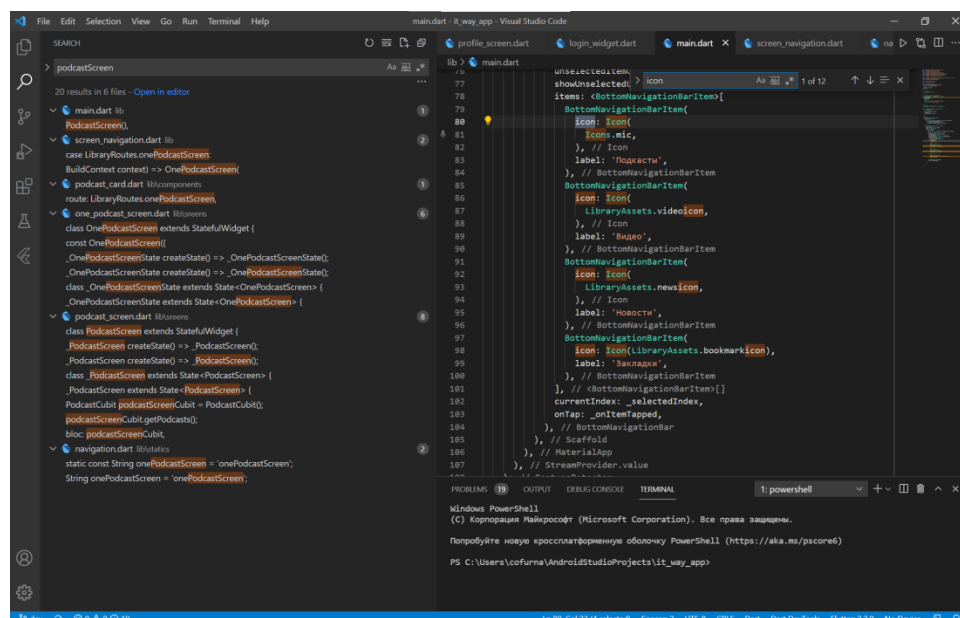


Рисунок 3 – Visual Studio Code.

Что касается Android Studio, то поиск перенаправляется в отдельное диалоговое окно, с точки зрения производительности это медленнее (из-за дополнительных действий).

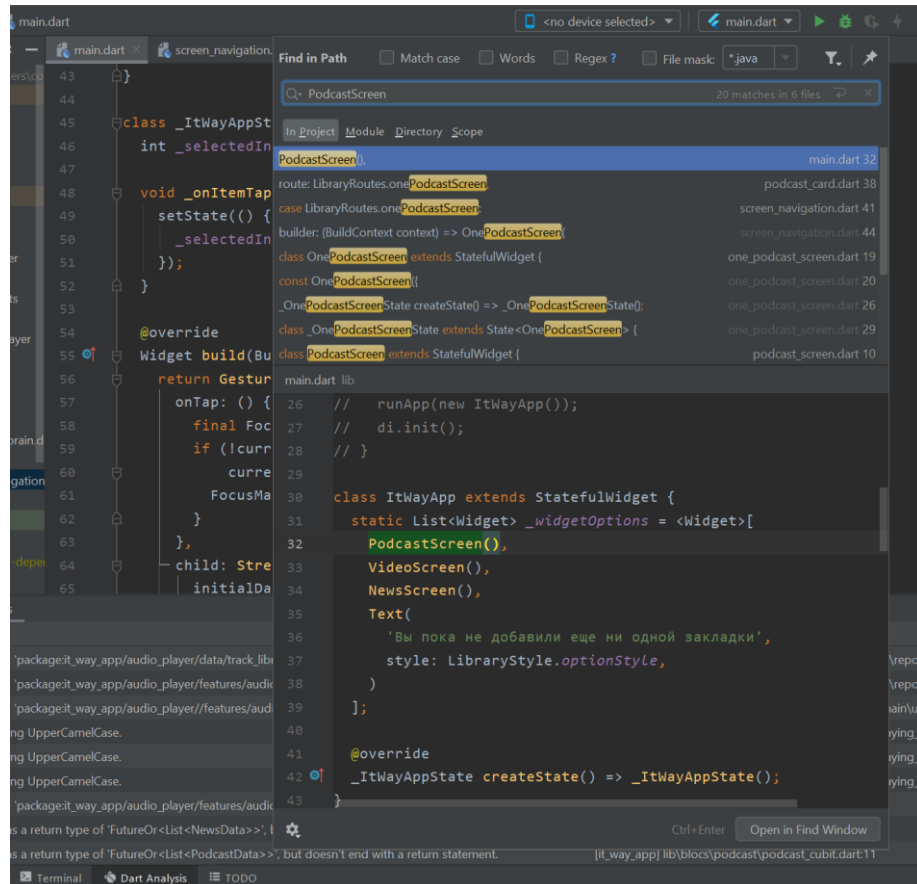


Рисунок 4 – Android Studio.

3.2.3 Контроль версий

Осуществить контроль версий проще в Visual Code, опять же по той причине, что не вызывается лишних диалоговых окон. Но в целом осуществляемая работа по добавлению изменений в проект схожа в обеих средах. Также для Visual Code существует множество плагинов, упрощающих написание кода.

3.3 Firebase

Для того чтобы разработать удачное мобильное приложение, необходимо несколько инструментов: хостинг (чтобы сохранять данные пользователя), коммуникация (для отправки на устройства уведомлений и различных рекламных акций), также очень важные инструменты – это тестирование и оптимизация, которые выявляют неполадки. В настоящее время, Google разработал платформу для решения этих нужд, которая называется Firebase. Firebase предлагает пользователю:

- 1 строить – развивать свой бэкэнд без управления серверами, обеспечить хранение и хостинг, облачные функции;
- 2 выпускать и мониторить – упрощать тестирование и устранить неполадки, следить за внедрением продукта, выявлять и расставлять приоритеты;
- 3 привлекать – повысить вовлеченность пользователей с помощью обширной аналитики, A / B-тестирования и рассылки сообщений.

Вместе с хранилищем, Firebase также предоставляет пользовательскую аутентификацию, и поэтому все данные передаются через защищенное соединение SSL. Возможно выбрать любую комбинацию email и пароля для аутентификации, будь то Facebook, Twitter, GitHub, Google, или что-то другое [6]. В данной работе будет выбран самый классический способ регистрации – ввод email и пароля, также будет использоваться платформа для хранения и мгновенного обновления комментариев с возможностью их оценивать.

На рисунке 5 представлены официальные стабильные плагины для Flutter.

Stable Plugins

Name	pub.dev	Firebase Product	Documentation	View Source	Mobile	Web	MacOS
Authentication				firebase_auth	✓	✓	✓
Cloud Firestore				cloud_firestore	✓	✓	✓
Cloud Functions				cloud_functions	✓	✓	✓
Cloud Messaging				firebase_messaging	✓	✓	✓
Cloud Storage				firebase_storage	✓	✓	✓
Core				firebase_core	✓	✓	✓
Crashlytics				firebase_crashlytics	✓	N/A	✓

Рисунок 5 – Плагины Firebase для Flutter.

3.4 Пакеты Dart и Flutter

Пакет – это код на Dart с pubspec.yaml файлом, а плагин – подвид пакета, который содержит нативный код какой-либо платформы [7]. Для того чтобы использовать пакет, сначала нужно добавить его в блок зависимостей, а далее импортировать в проект, на рисунке 6 представлено добавление в текущий проект, и последующий импорт.

```
dependencies:
  flutter:
    sdk: flutter
  http: ^0.13.3

  json_serializable: ^4.1.3
  bloc: ^7.0.0
  flutter_bloc: ^7.0.0
  audioplayers: ^0.18.3
  audio_service: ^0.17.0
  get_it: ^6.0.0
  flutter_svg: ^0.21.0-nullsafety.0
  youtube_player_flutter: ^8.0.0
  firebase_auth: ^1.2.0
  cloud_firestore: ^2.2.0
  provider: ^5.0.0
  fluttertoast: ^8.0.7
  firebase_core: ^1.2.0

import 'package:audio_service/audio_service.dart';
import 'package:audioplayers/audioplayers.dart';
```

Рисунок 6 – Добавление и использование пакетов.

3.4.1 Firebase_auth, provider, firebase_core

Firestore_auth – пакет, который упрощает создание безопасных систем аутентификации для конечных пользователей и разработчиков. Он поддерживает различные решения регистрации и входа. Информация пользователя содержит уникальный идентификатор, что позволяет персонализировать пользовательский интерфейс для конкретного пользователя. На рисунке 7 показан результат использования пакета.



Рисунок 7 – Firestore_auth

Firestore_core – плагин, который отвечает за подключение приложения к проекту Firebase. Для использования служб Firebase сначала нужно инициализировать FlutterFire.

```
Run | Debug
Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();

  runApp(new ItWayApp());
}
```

Рисунок 8 – Метод инициализации.

Provider – пакет для управления состоянием приложения. Например, в проекте на самом начале – это точка входа в приложение, в которой нужно зарегистрировать состояние, необходимое для передачи всему приложению. На рисунке 9 видно, как это сделать. Нужно это для отображения либо страницы пользователя, либо входа/регистрации.

```

child: StreamProvider<UserProfile>.value(
  initialData: null,
  value: AuthService().currentUser,
  child: new MaterialApp(
    home: Scaffold(
      body: Center(
        child: ItWayApp._widgetOptions.elementAt(_selectedIndex),

```

Рисунок 9 – Прослушивание состояния.

3.4.2 Bloc

Bloc – компонент бизнес логики, шаблон, созданный Google для управления сложным состоянием приложения, он отделяет пользовательский интерфейс от бизнес логики. Слой блока принимает какое-то событие, которое сгенерировано пользователем, и возвращает состояние для уровня представления в зависимости от события. Получается, что ответственность уровня блока заключается в том, чтобы отвечать на события из уровня представления новыми состояниями, а ответственность уровня представления в том, чтобы выяснить, как визуализировать себя на основе одного или нескольких состояний, которые вернет блок. Если же приложение получает данные по сети или от других асинхронных источников данных, то добавляется третий слой – слой данных – самый низкий уровень приложения. На рисунке 10 изображен шаблон блока [8].

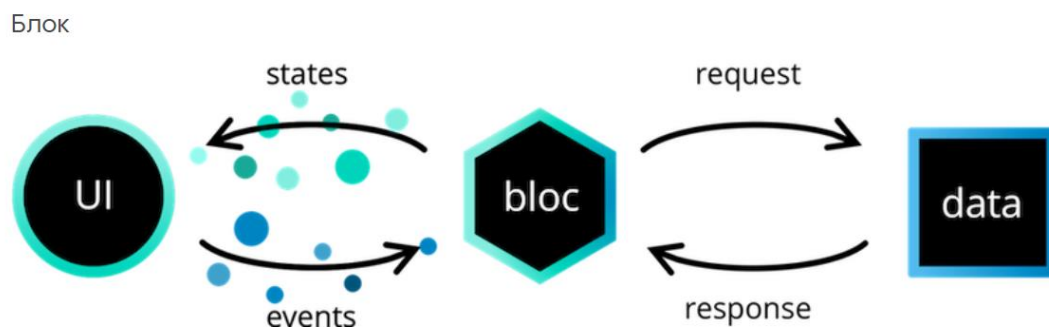


Рисунок 10 – шаблон Bloc.

Также стоит обратить внимание на Cubit. Cubit – это более легкая версия блока, на котором основан пакет flutter-bloc. Вместо событий здесь используется функции, единственное отличие – это реализация вызовов состояний, через метод emit. Cubit требует начального состояния, которое будет состоянием до того, emit было вызвано. На рисунке 11 изображен шаблон cubit [8].

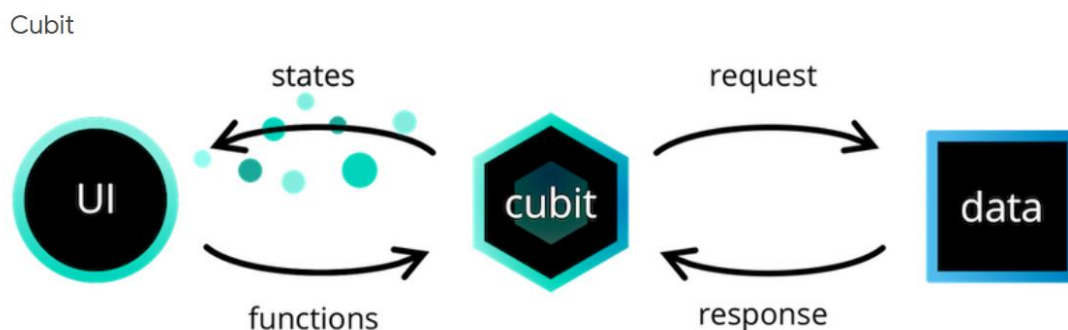


Рисунок 11 – шаблон Cubit.

3.4.3 Audioplayers, audio_service

Audioplayers – одна из лучших библиотек для воспроизведения звуковых файлов, в основном нацелена на игры. Обеспечивает минимальную задержку при воспроизведении звука. Из минусов – нет медиа-уведомлений. Эту проблему решает пакет Audio_service.

Audio_service – библиотека, позволяющая запускать аудио в фоне и контролировать его различными способами (рисунок 12).

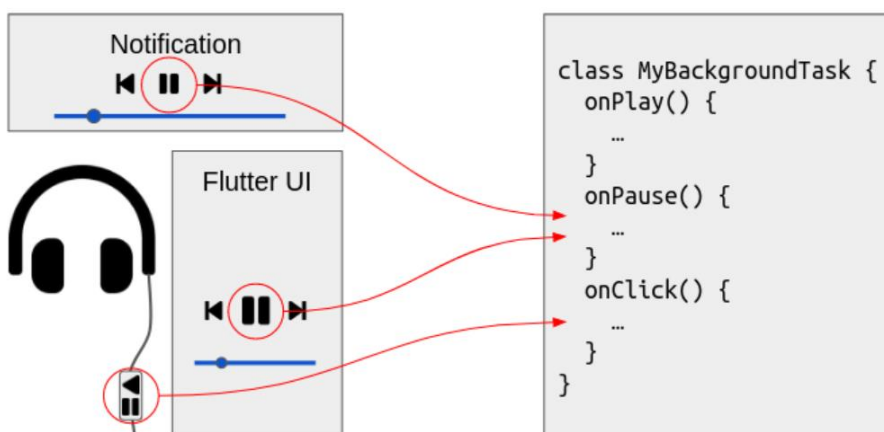


Рисунок 12 – audio_service.

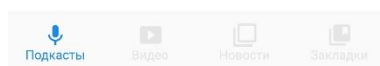
3.4.4 Youtube_player_flutter

Youtube_player_flutter – пакет для воспроизведения видео с YouTube. Существенная особенность в том, что пакет поддерживает все функции плеера, такие как: субтитры, изменение скорости воспроизведения, полноэкранный режим, быстрая перемотка, встроенное воспроизведение, поддержка видео в прямом эфире, адаптация к качеству в зависимости от пропускной способности.

4 Результаты проведенной разработки

В ходе выполнения работы была написана программа, представляющая собой мобильное приложение под ОС Android. Далее приводится демонстрация всех реализованных функций, начиная с запуска приложения.

На рисунке 13а показан виджет загрузки, который состоит из названия приложения и круглого индикатора загрузки. Данный виджет отображается на каждой странице разного контекста, где визуальное представление зависит от статуса состояния кубита. Для примера, на рисунке 13б располагается часть кода страницы подкастов.



а)

```
child: BlocBuilder<PodcastCubit, PodcastStates>(
  bloc: podcastScreenCubit,
  builder: (BuildContext context, PodcastStates state) {
    if (state.status == PodcastStatus.dataLoading) {
      return LoadingApp();
    }
  }
)
```

б)

Рисунок 13 – а) индикатор загрузки экранов, б) часть кода

Первая страница на которой оказывается пользователь – страница подкастов (рисунок 14).

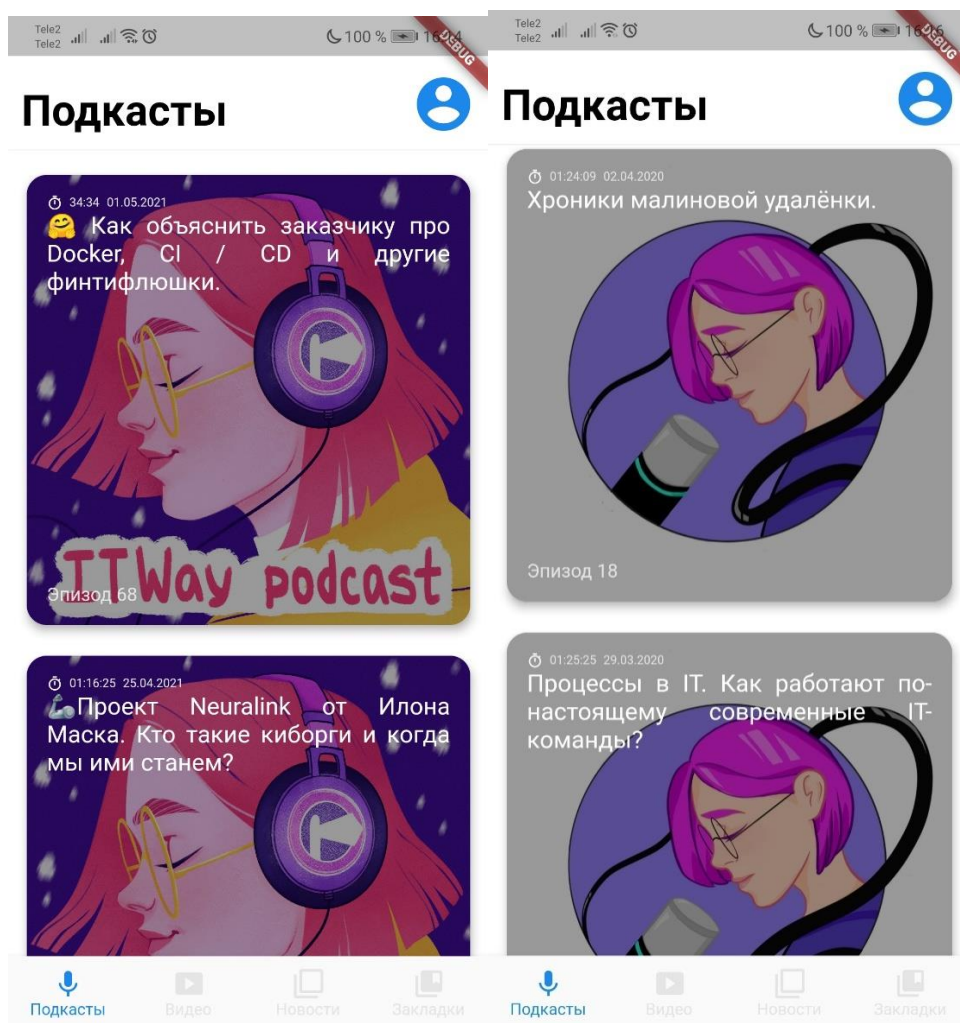


Рисунок 14 – Страница подкастов.

Как и любая другая страница с различным контентом, она представляет список карточек, в каждую из которых по нажатию передаются данные, и осуществляется переход на страницу непосредственно самого контента. В нашем случае это аудиоплеер, который позволяет прослушивать подкасты (реализация отображена на рисунке 15). Также видно, что у пользователя есть возможность перематывать подкаст с помощью скроллбара.

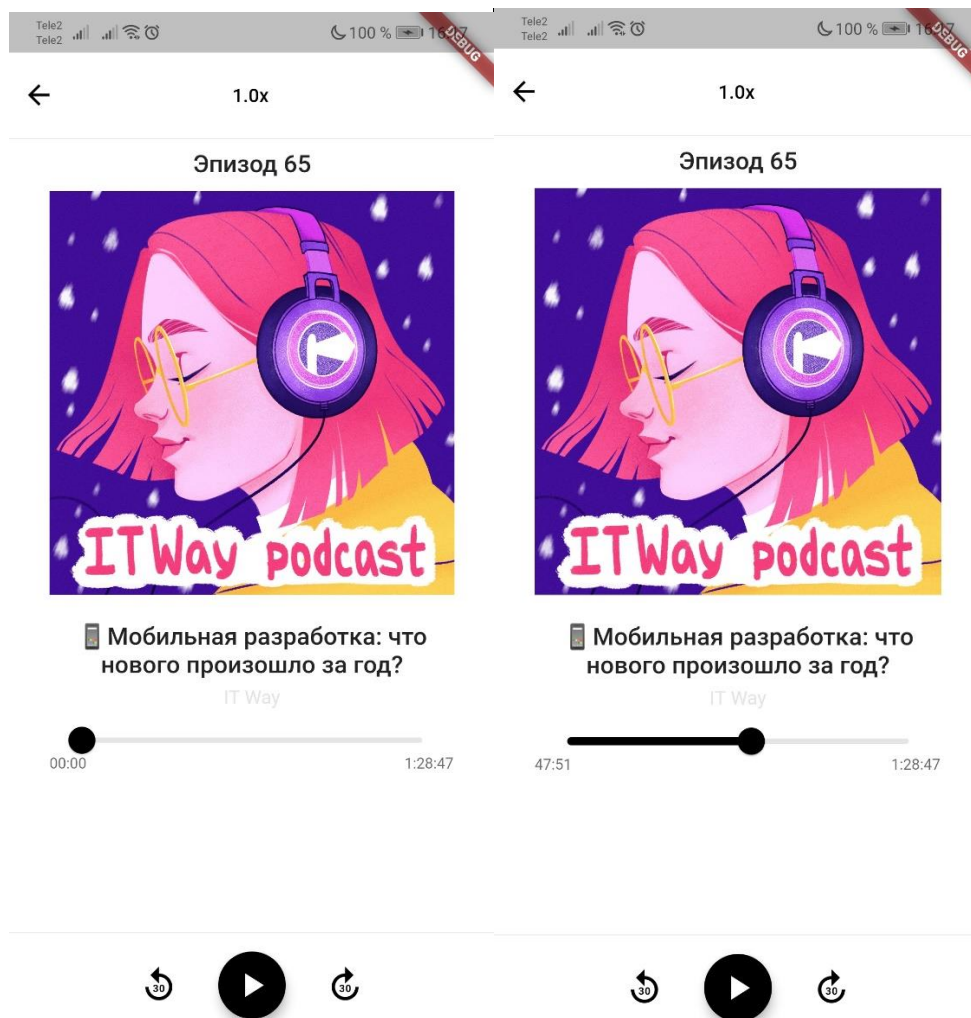


Рисунок 15 – Подкаст-плеер.

На рисунке 16 представлена перемотка на 30 секунд назад и на 30 секунд вперед, осуществляемая нажатием на ту или иную кнопку соответственно. Для реализации плеера используются пакеты аудиоблока, которые были описаны ранее.

Также основой для плеера является объект `duration` (продолжительность). В `appBar` располагается кнопка, позволяющая контролировать скорость воспроизведения. На рисунке 17 показано уведомление, в котором можно остановить, либо воспроизвести подкаст, а также удалить его из фоновой службы (изолировать).

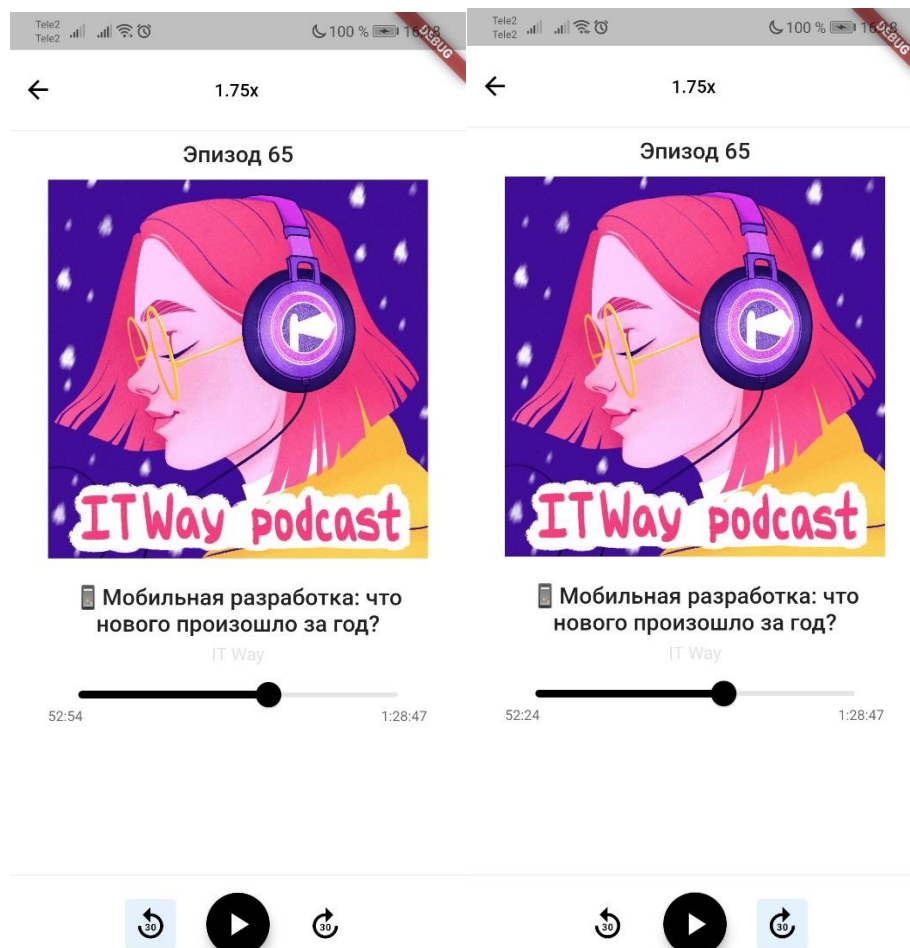


Рисунок 16 – Перемотка на 30 секунд.

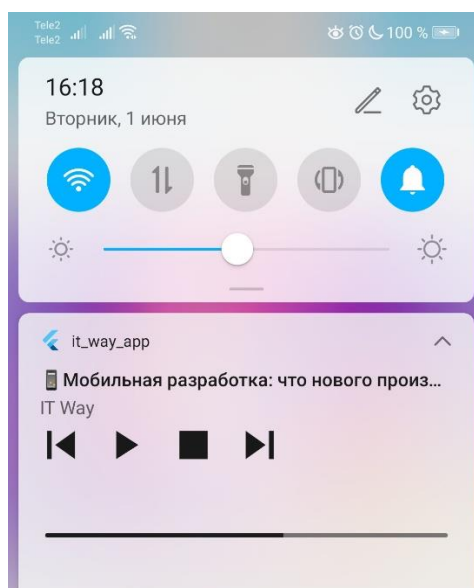
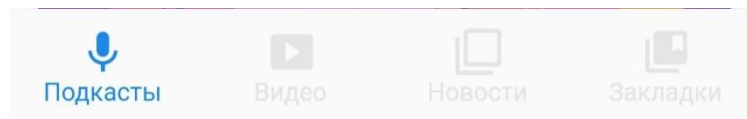


Рисунок 17 – Уведомление о воспроизведении подкаста.

Для перемещения по страницам был написан код, представленный на рисунке 18а, рисунок 18б представляет собой панель нижней навигации (bottomNavigationBar).

```
67 child: new MaterialApp(  
68   home: Scaffold(  
69     body: Center(  
70       child: ItWayApp._widgetOptions.elementAt(_selectedIndex),  
71     ), // Center  
72     bottomNavigationBar: BottomNavigationBar(  
73       type: BottomNavigationBarType.fixed,  
74       selectedFontSize: 12.0,  
75       selectedItemColor: LibraryColors.kActiveColor,  
76       unselectedItemColor: LibraryColors.kUnActiveColor,  
77       showUnselectedLabels: true,  
78       items: <BottomNavigationBarItem>[  
79         BottomNavigationBarItem(  
80           icon: Icon(  
81             Icons.mic,  
82           ), // Icon  
83           label: 'Подкасты',  
84         ), // BottomNavigationBarItem  
85         BottomNavigationBarItem(  
86           icon: Icon(  
87             LibraryAssets.videoicon,  
88           ), // Icon  
89           label: 'Видео',  
90         ), // BottomNavigationBarItem  
91         BottomNavigationBarItem(  
92           icon: Icon(  
93             LibraryAssets.newsicon,  
94           ), // Icon  
95           label: 'Новости',  
96         ), // BottomNavigationBarItem  
97         BottomNavigationBarItem(  
98           icon: Icon(LibraryAssets.bookmarkicon),  
99           label: 'Закладки',  
100        ), // BottomNavigationBarItem  
101      ], // <BottomNavigationBarItem>[]  
102      currentIndex: _selectedIndex,  
103      onTap: _onItemTapped,  
104    ), // BottomNavigationBar  
105  ), // Scaffold  
106 ), // MaterialApp
```

а)



б)

Рисунок 18 – панель нижней навигации.

На рисунке 19 представлено воспроизведение видео. Данные видео (так же, как для подкастов и новостей) получаются через API и парсятся. Пакет видеоплеера реализован в более простом виде, чем пакет аудио, и он предоставляет все функции, описанные ранее, которые показаны на рисунке 20.

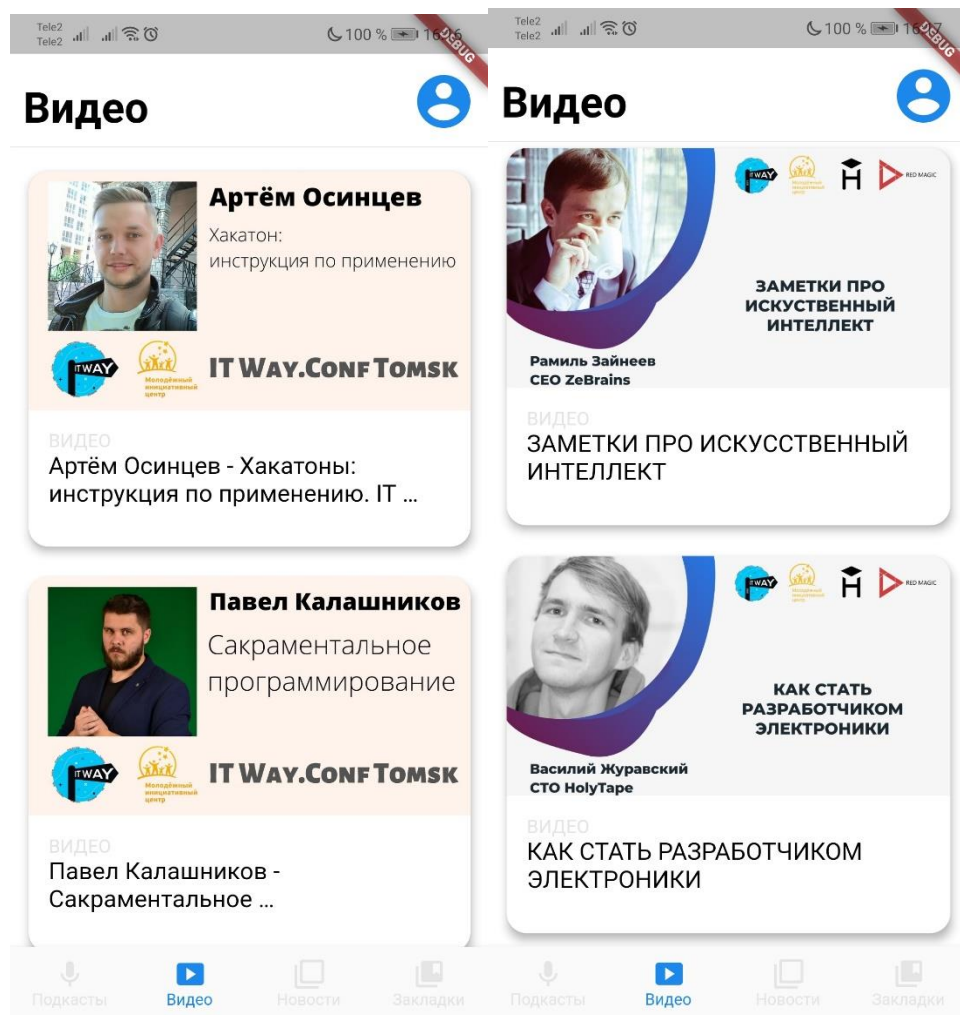
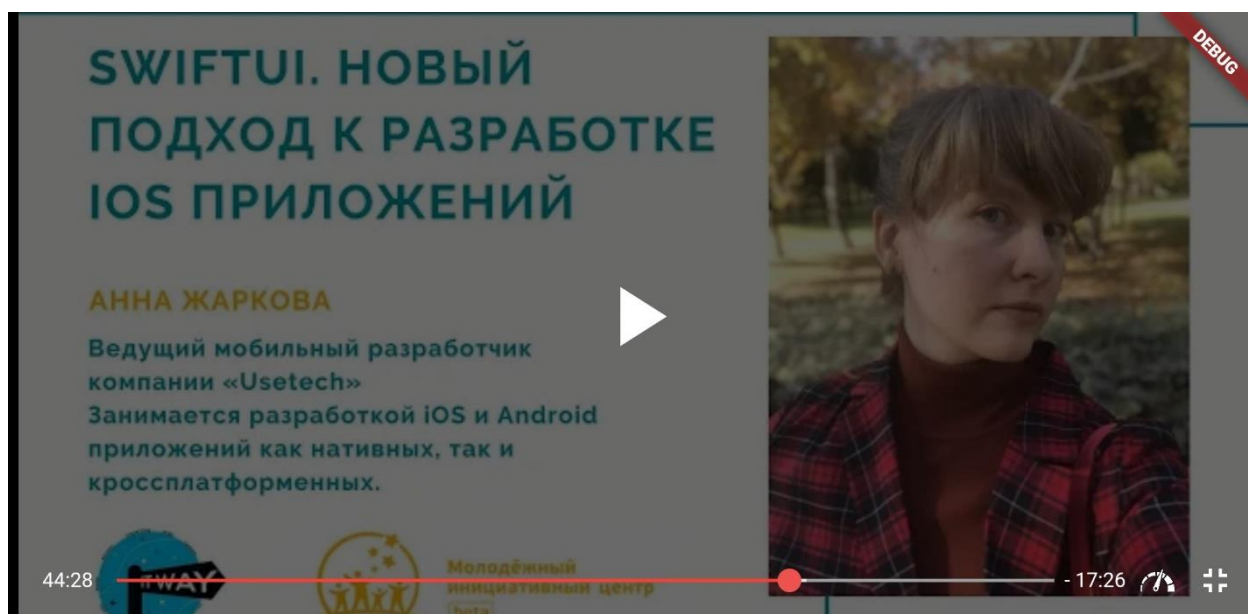
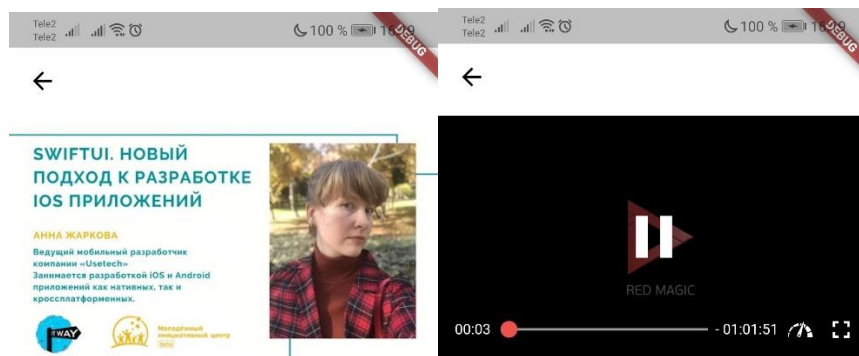


Рисунок 19 – Страница подкастов.



а)



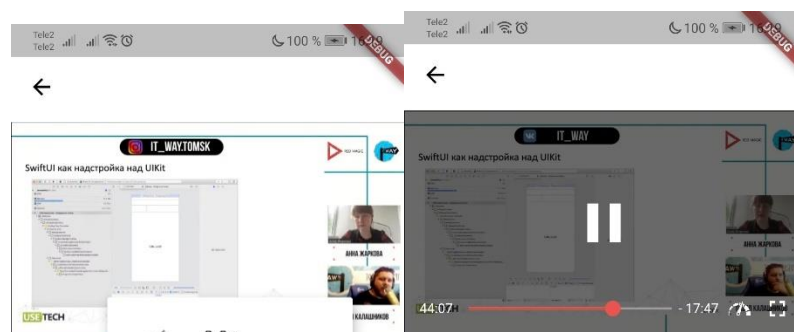
SwiftUI - новый подход к разработке iOS приложений. IT Way.Tomsk. 31.07.2020

- Анна Жаркова — ведущий мобильный разработчик компании «Usetech» с опытом коммерческой разработки 7 лет. Занимается разработкой iOS и Android приложений как нативных (Swift/Objective-C, Kotlin/Java), так и кроссплатформенных (Xamarin iOS/Android, Xamarin Forms, Kotlin Multiplatform). Разрабатывает архитектуру мобильных приложений для обеих платформ, управляет командой направления, ведет младших разработчиков (менторство).

SwiftUI - новый подход к разработке iOS приложений. IT Way.Tomsk. 31.07.2020

- Анна Жаркова — ведущий мобильный разработчик компании «Usetech» с опытом коммерческой разработки 7 лет. Занимается разработкой iOS и Android приложений как нативных (Swift/Objective-C, Kotlin/Java), так и кроссплатформенных (Xamarin iOS/Android, Xamarin Forms, Kotlin Multiplatform). Разрабатывает архитектуру мобильных приложений для обеих платформ, управляет командой направления, ведет младших разработчиков (менторство).

б)



SwiftUI как надстройка над UIKit

- Анна Жаркова — ведущий мобильный разработчик компании «Usetech» с опытом коммерческой разработки 7 лет. Занимается разработкой iOS и Android приложений как нативных (Swift/Objective-C, Kotlin/Java), так и кроссплатформенных (Xamarin iOS/Android, Xamarin Forms, Kotlin Multiplatform). Разрабатывает архитектуру мобильных приложений для обеих платформ, управляет командой направления, ведет младших разработчиков (менторство).
- Среди ее проектов приложения для банков и служб занятости, для транспортных компаний, вело- и каршеринга. Также Анна разрабатывает справочные приложения и программы лояльности

SwiftUI - новый подход к разработке iOS приложений. IT Way.Tomsk. 31.07.2020

- Анна Жаркова — ведущий мобильный разработчик компании «Usetech» с опытом коммерческой разработки 7 лет. Занимается разработкой iOS и Android приложений как нативных (Swift/Objective-C, Kotlin/Java), так и кроссплатформенных (Xamarin iOS/Android, Xamarin Forms, Kotlin Multiplatform). Разрабатывает архитектуру мобильных приложений для обеих платформ, управляет командой направления, ведет младших разработчиков (менторство).

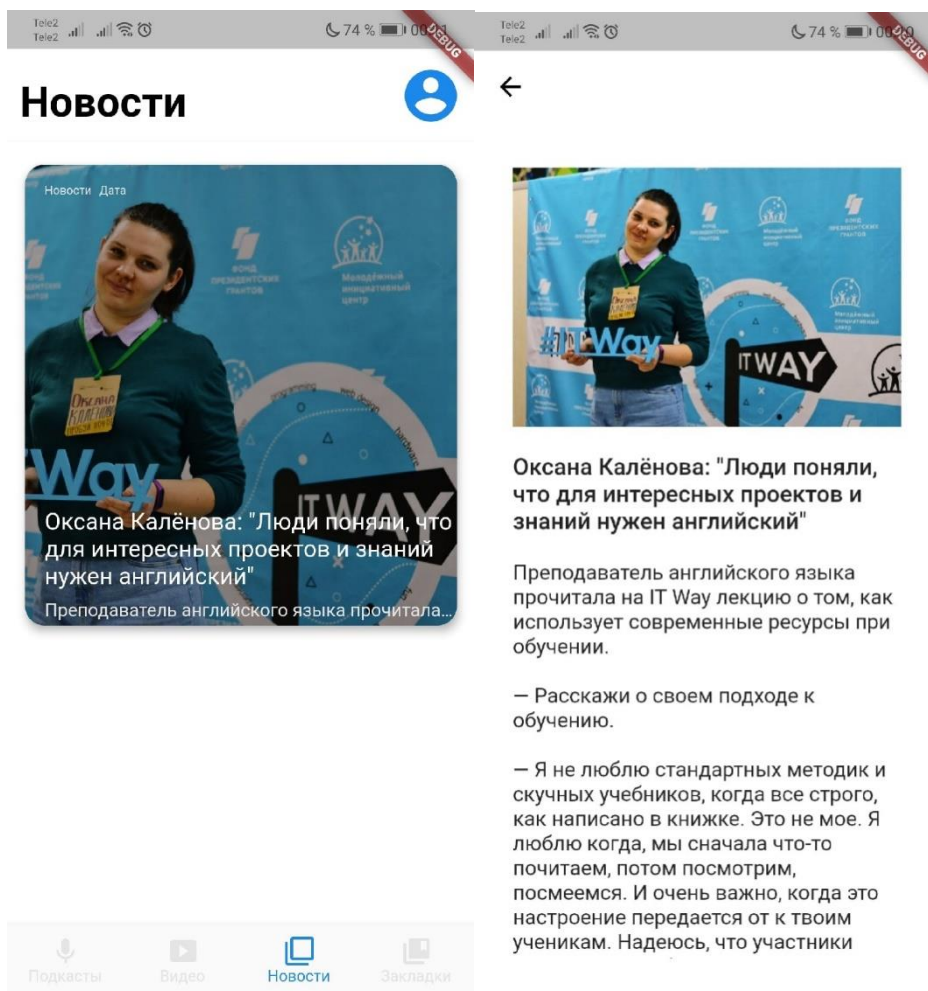
- Среди ее проектов приложения для банков и служб занятости, для транспортных компаний, вело- и каршеринга. Также Анна разрабатывает справочные приложения и программы лояльности

в)

г)

Рисунок 20 – а) полноэкранный режим, б) страница одного видео, в) изменение скорости, г) перемотка видео с помощью скроллбара

В API новостей пока находится только одна новость, она показана на рисунке 21.



а)

б)

Рисунок 21 – а) страница новостей б) одна новость

На каждой странице находится панель действий (так называемая AppBar). На рисунке 22 приведен код двух таких панелей. Основная —определяет страницу и имеет виджет перехода к странице регистрации/входа или странице пользователя. Вторая — содержит кнопку перехода назад и используется на странице одиночного контента.



а)



б)

Рисунок 22 – а) AppBar б) AppBarBack.

Виджетом для перехода к странице профиля является круглый значок аватара. В приложении будет доступна классическая регистрация с вводом логина и почты (рисунок 23).

Страница пользователя создается с заданными именем (рисунок 24). На ней располагается кнопка выхода.

Также на этих страницах доступны переход на сайт сообщества и переход к чату в телеграмме. Все пользователи сохраняются в консоли Firebase (рисунок 25).

Листинг основных файлов программы представлен в приложении А.

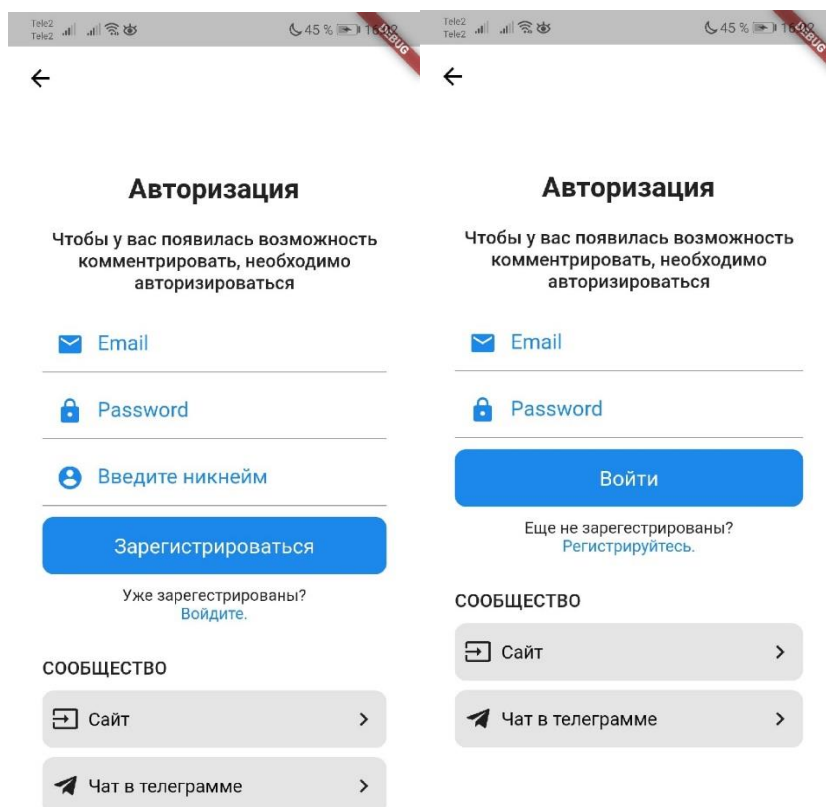


Рисунок 23 – Регистрация и вход.

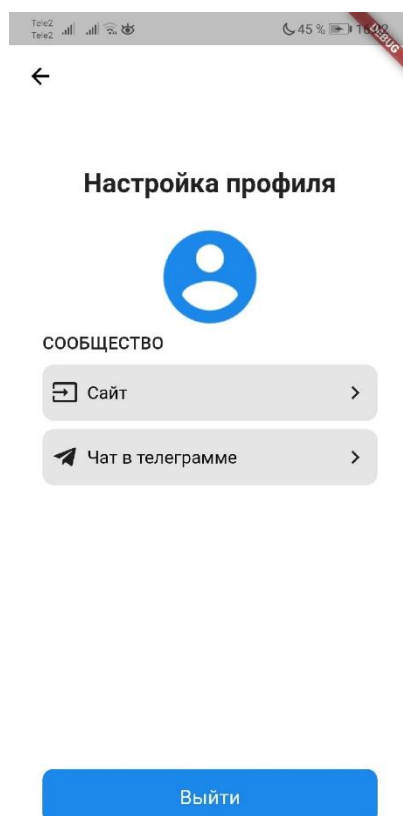


Рисунок 24 – Страница одного пользователя

Search by email address, phone number or user UID					Add user	↺	⋮
Identifier	Providers	Created	Signed in	User UID	↑		
andranna567@gmail.com	✉	30 May 2021	1 Jun 2021	Ifudr4cbkRZKnckARFFzNNVX0NH2			
stakan335@gmail.com	✉	31 May 2021	31 May 2021	QpxRnliM2ZYIZRuYPpYFb2UejJq1			
asa48@tpu.ru	✉	1 Jun 2021	1 Jun 2021	b4BOs5rUxGgt2urPseGbJORS7IN2			
					Rows per page	50	1 – 3 of 3

itwayapp	user	wjiJYQmXnsh7xnSDCsIcjKGyqr82
+ Start collection	+ Add document	+ Start collection
user	QSpCL0otlgZg7Zr9VC8EeEZsha02 RU2n1QYb4FRTfAhWyUjpSfmUHL62 user wjiJYQmXnsh7xnSDCsIcjKGyqr82	+ Add field name: "Анна Андросова"

Рисунок 25 – Зарегистрированные пользователи.

Также все изменения проекта записываются в репозиторий GitHub в ветку dev (рисунок 26).

dev

2 branches

0 tags

Go to file

Add file

Code

This branch is 21 commits ahead of master.

Pull request

Compare

cofurna debugging

35a097b 3 hours ago 25 commits

.vscode	auth screen implemented, video screen implemented,icons made	6 days ago
android	UserScreen? navigation are implemented	2 days ago
assets	auth screen implemented, video screen implemented,icons made	6 days ago
ios	Initial commit	10 months ago
lib	debugging	3 hours ago
test	add podcasts	25 days ago
.gitignore	Initial commit	10 months ago
.metadata	Initial commit	10 months ago
README.md	Initial commit	10 months ago
pubspec.lock	UserScreen? navigation are implemented	2 days ago
pubspec.yaml	UserScreen? navigation are implemented	2 days ago

Рисунок 26 – репозиторий на GitHub.

5 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

5.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

5.1.1 Потенциальные потребители результатов исследования

Согласно данным исследовательской компании Canalys, мировые поставки смартфонов достигли 347 миллионов единиц, что на 27% больше, чем в прошлом году [2]. При этом больше половины управляются посредством операционной системы Android. Также доля рынка, занимаемая данной ОС в России, составляет 75,35% [9]. Из чего можно сделать вывод, что мобильные устройства активно используются, соответственно и разнообразные приложения для смартфонов тоже.

Мобильное приложение предназначено для всех участников сообщества «IT Way» и для людей, интересующихся IT, обеспечивает комфортное потребления различной информации, такой как видео и подкасты, статьи и новости.

5.1.2 Анализ конкурентных технических решений

Для того, чтобы узнать, насколько проект будет эффективен на рынке, следует провести анализ конкурентных технических решений, также это поможет в направленности приложения на развитие.

Перейдем к анализу. Позиция технического решения и конкурентов оценивается по каждому показателю в пятибалльной шкале, где 5 является сильной позицией, а 1 наиболее слабая. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1.

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum B_i \cdot \text{Б}_i, \quad (1)$$

где K – конкурентоспособность научной разработки или конкурента;

B_i – вес показателя (в долях единицы);

B_i – балл i -го показателя.

В Таблице 2 приведена оценочная карта для сравнения конкурентных мобильных приложений. Стоит отметить, что ни одно приложение не является прямым конкурентом. Первым конкурентом выберем приложение «Wylsacom» – программа аналогичная по своей структуре, но отличная по содержанию; вторым – «Meduza», а третьим – «Яндекс.Дзен – интересные статьи, видео и новости». Последние два конкурента уступают в разнообразии видов контента.

Таблица 2 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы				Конкурентоспособность			
		B_{ϕ}	B_{K1}	B_{K2}	B_{K3}	K_{ϕ}	K_{K1}	K_{K2}	K_{K3}
1	2	3	4	5	6	7	8	9	10
Технические критерии оценки ресурсоэффективности									
1. Удобство в эксплуатации	0,15	5	5	4	3	0,75	0,75	0,6	0,45
2. Функциональная мощность	0,25	5	5	4	4	1,25	1,25	1	1
3. Простота эксплуатации	0,1	5	5	5	4	0,5	0,5	0,5	0,4
4. Надежность	0,2	5	5	5	3	1	1	1	0,6
Экономические критерии оценки эффективности									
1. Конкурентоспособность продукта	0,1	4	4	4	3	0,4	0,4	0,4	0,3
2. Послепродажное сопровождение	0,1	5	5	5	5	0,5	0,5	0,5	0,5
3. Предполагаемый срок эксплуатации	0,1	5	5	5	5	0,5	0,5	0,5	0,5

Итого	1					4,9	4,9	4,5	3,75
--------------	----------	--	--	--	--	-----	-----	-----	------

Взяв за основу полученные результаты конкурентных технических решений, можно сделать вывод, что разработка обладает конкурентным преимуществом в двух случаях из трех, подкрепляя его мощным функционалом. Что касается первого конкурента, то здесь речь идет о следующем: оба приложения представляют собой площадку с доступом к информации (имеет разные типы отображение – текст, видео, аудио), ориентированной на конкретный круг пользователей.

5.1.3 Анализ по технологии QuaD

Одной из технологий для оценки качественных характеристик является технология QuaD. Каждый показатель оценивается экспертным путем по столбальной шкале, где 1 – наиболее слабая позиция, а 100 – наиболее сильная. Веса показателей, определяемые экспертным путем, в сумме должны составлять 1. В таблице 3 представлена оценка конкурентных технических решений.

Оценка качества и перспективности по технологии QuaD определяется по формуле:

$$P_{\text{ср}} = \sum B_i \cdot \bar{B}_i, \quad (2)$$

где $P_{\text{ср}}$ – средневзвешенное значение показателя качества и перспективности научной разработки;

B_i – вес показателя (в долях единицы);

\bar{B}_i – средневзвешенное значение i -го показателя.

Таблица 3 – Оценочная карта для сравнения конкурентных технических решений по технологии QuaD

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение (3/4)	Среднее взвешенное значение (5x2)
1	2	3	4	5	6
Показатели оценки качества разработки					

1. Функциональная мощность	0,25	90	100	0.9	0,225
2. Качество визуального оформления	0,1	80	100	0,8	0,08
3. Простота эксплуатации	0,15	100	100	1	0,15
4. Надежность	0,15	90	100	0,9	0,135
Показатели оценки коммерческого потенциала разработки					
1. Конкурентно способность продукта	0,1	80	100	0,8	0,08
2. Перспективность рынка	0,15	60	100	0,6	0,09
3. Послепродажное обслуживание	0,1	100	100	1	0,1
Итого	1				0,860

Значение показателя Пср составляет 86%, из чего можно сделать вывод, разработка перспективна.

5.1.4 SWOT-анализ

В ходе проведения SWOT анализа были выявлены сильные и слабые стороны проекта, а также его возможности и угрозы. Результат SWOT-анализа представлен в Таблице 4.

Таблица 4 – матрица SWAT

	Сильные стороны проекта: С1. Отсутствие прямых конкурентов; С2. Функциональность; С3. Простой интерфейс.	Слабые стороны проекта: Сл1. Необходимость подключения к сети интернет; Сл2. Отсутствие опыта разработки.
Возможности: В1. Облегчение процесса получения контента; В2. Масштабируемость.	Разработка приложения востребована, в ходе масштабирования функционала повышается спрос.	С развитием проекта опыт будет прибавляться.

Угрозы: У1. Возможная не востребованность.	Сообщество немногочисленно, соответственно мало людей может пользоваться приложением.	Возможность сделать невостребованное приложение, за счет малого опыта разработки.
--	---	---

Ориентируясь на результаты SWOT-анализа, можно сказать, что проект выгоден, однако нужно направить все силы на качественную разработку продукта.

5.2 Определение возможных альтернатив проведения научных исследований

Для того, чтобы определить альтернативные пути проведения научного исследования и варианты реализации работы, используется морфологическая матрица, которая представлена в таблице 5.

Таблица 5 – Морфологическая матрица

	1	2
А. Используемый язык	Kotlin/Java, Swift	Dart (Flutter)

Выбор используемого языка основывается на знаниях студента, а также последующей возможности расширения проектирования под IOS. Из полученной морфологической матрицы, можно получить 2 варианта реализации проекта:

- Исполнение 1 – А1;
- Исполнение 2 – А2.

5.3 Планирование научно-исследовательских работ

5.3.1 Структура работ в рамках научного исследования

Важным этапом проведения научно-исследовательских работ является необходимость планирования комплекса предполагаемых работ: определение структуры, участников, продолжительности. Исполнителями проекта являются студент и научный руководитель. В Таблице 6 представлен перечень этапов и работ, а также распределение исполнителей по данным видам работ в рамках проводимого научно-исследовательского проекта.

Таблица 6 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания	Руководитель, инженер
	2	Календарное планирование работ по теме	Руководитель, инженер
Аналитический обзор	3	Подбор и изучение литературы по теме	Инженер
	4	Выбор инструментов разработки	Инженер
Реализация и тестирование	7	Разработка мобильного приложения	Инженер
	9	Тестирование приложения	Инженер
Оценка результатов и оформление отчетности	10	Согласование выполненной работы	Руководитель, инженер
	11	Оценка результатов	Руководитель
	12	Оформление пояснительной записки и необходимой документации	Руководитель, инженер

5.3.2 Определение трудоемкости выполнения работ

Далее стоит определить трудоемкость работ каждого участника проекта, т.к. трудовые затраты закладывают основную часть разработки.

Для определения ожидаемого значения трудоемкости тоже используется следующая формула:

$$t_{ож\ i} = \frac{3t_{mini} + 2t_{maxi}}{5}, \quad (3)$$

где $t_{ож\ i}$ – ожидаемая трудоемкость выполнения i -ой работы на чел.-дн.;

t_{mini} – минимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.;

t_{maxi} – максимально возможная трудоемкость выполнения заданной i -ой работы, чел.-дн.

По формуле, представленной ниже, исходя из ожидаемой трудоемкости работ определяется продолжительность каждой работы в рабочих днях T_p , учитывающая параллельность выполнения работ несколькими исполнителями:

$$T_{p\ i} = \frac{t_{ож\ i}}{ч_i}, \quad (4)$$

где $T_{p\ i}$ – продолжительность одной работы, раб. дн.;

$t_{ож\ i}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.;

$ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

5.3.3 Разработка графика проведения научного исследования

Для наглядности и удобства восприятие проведенной работы построим горизонтальный ленточный график в форме диаграммы Ганта. Перевод длительности каждого из этапов работ из рабочих дней в календарные дни ведется по формуле:

$$T_{k\ i} = T_{p\ i} \cdot k_{кал}, \quad (5)$$

где $T_{k\ i}$ – продолжительность выполнения i -й работы в календарных днях;

$T_{p\ i}$ – продолжительность выполнения i -й работы в рабочих днях;

$k_{кал}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{кал} = \frac{T_{кал}}{T_{кал} - T_{вых} - T_{пр}}, \quad (6)$$

где $T_{кал}$ – количество календарных дней в году;

$T_{вых}$ – количество выходных дней в году;

$T_{\text{пр}}$ – количество праздничных дней в году.

Согласно производственному календарю (для шестидневной рабочей недели) в 2021 году 365 календарных дней, из них 66 дней – выходные и праздничные дни. Таким образом, согласно формуле 6 коэффициент календарности равен:

$$k_{\text{кал}} = \frac{365}{365 - 66} = 1,22$$

Временные показатели научного исследования, рассчитанные нами, сведены в таблицу 7, где Р – руководитель, И – инженер. На основе таблицы построен календарный план-график исследований, который представлен на рисунке 27 (за основу берутся максимальные значения дней для каждого пункта).

Таблица 7 – Временные показатели научного исследования

№	Название работы	Трудоемкость работ, чел-дни						Загрузка исполнителей	Длительность работ			
		tmin		tmax		toж			Тpi		Тki	
		Ипс1	Исп 2	Исп1	Исп2	Исп1	Исп2		Исп1	Исп2	Исп1	Исп2
1	Составление и утверждение технического задания	4	4	6	6	4,8	4,8	Р – 20%	0,96	0,96	1	1
								И – 80%	3,84	3,84	5	5
2	Календарное планирование работ по теме	2	2	4	4	2,8	2,8	Р – 50%	1,4	1,4	2	2
								И – 50%	1,4	1,4	2	2
3	Подбор и изучение литературы по теме	20	10	30	20	24	14	И – 100%	24	14	29	17
4	Выбор инструментов разработки	1	1	2	2	1,4	1,4	И – 100%	1,4	1,4	2	2
5	Разработка мобильного приложения	55	30	65	40	59	34	И – 100%	59	34	72	41
6	Тестирование приложения	12	10	17	15	14	12	И – 100%	14	12	17	15
7	Согласование выполненной работы	2	2	3	3	2,4	2,4	Р – 50%	1,2	1,2	1	1
								И – 50%	1,2	1,2	1	1
8	Оценка результатов	2	2	4	4	2,8	2,8	Р – 100%	2,8	2,8	3	3
9	Оформление пояснительной записки и необходимой документации	5	5	12	12	7,8	7,8	И – 100%	7,8	7,8	10	10
Итого		Руководитель						6,36	6,36	7	7	
		Инженер						112,64	75,64	138	93	

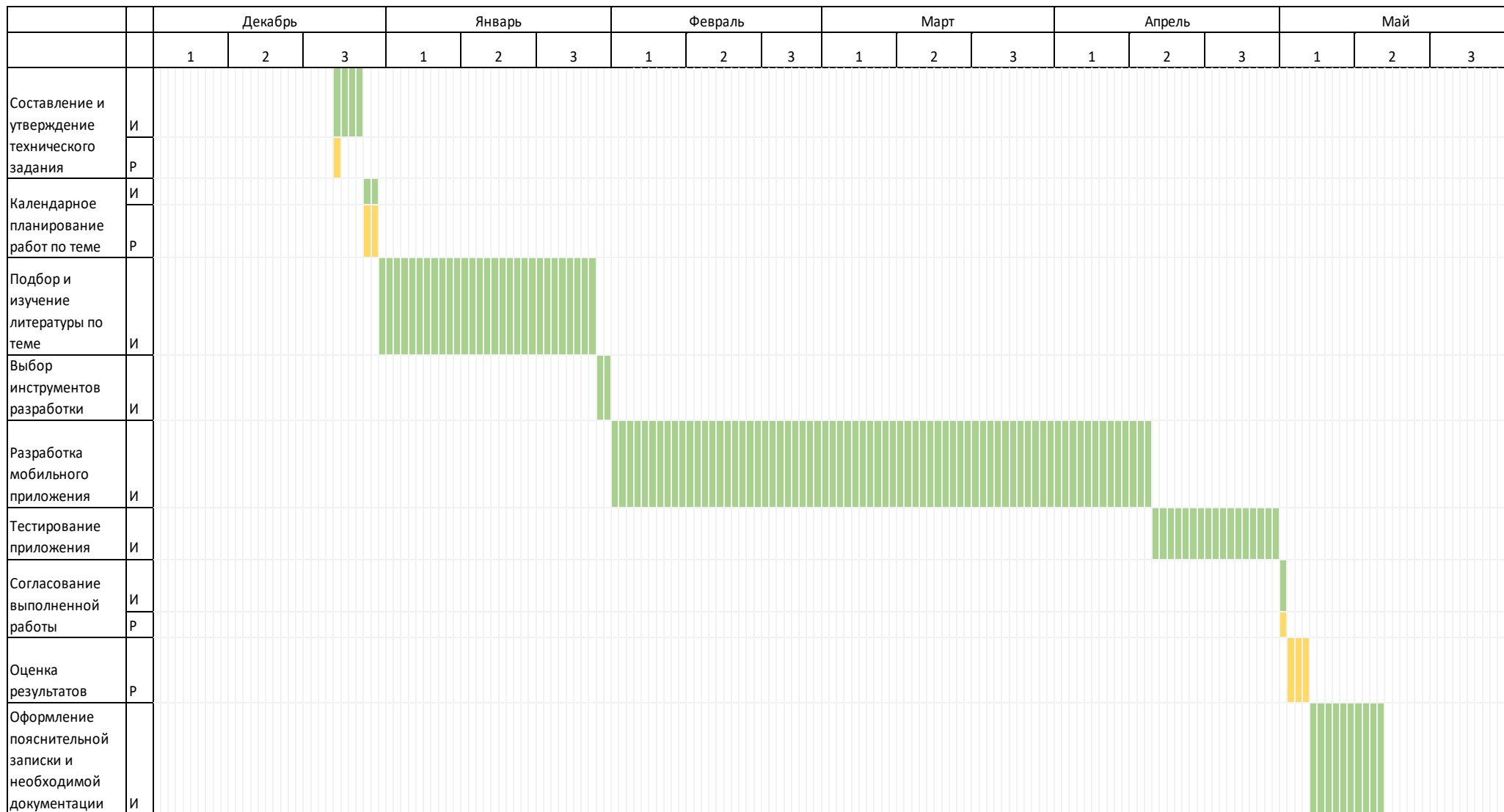


Рисунок 27 – Линейный график работы

5.3.4 Бюджет научно-технического исследования (НТИ)

При планировании бюджета НТИ должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. Для расчета стоимости выполнения проекта используются следующие статьи затрат:

- материальные затраты НТИ;
- заработная плата исполнителей;
- отчисления во внебюджетные фонды;
- расходы на электроэнергию (без освещения);
- амортизационные расходы;
- накладные расходы.

5.3.4.1 Расчет материальных затрат НТИ

Материальные затраты включают в себя: сырье и материалы, покупные материалы для производственных и хозяйственных нужд, затраты на канцелярию. В процессе выполнения ВКР дополнительных материалов не приобреталось.

5.3.4.2 Расчет затрат на специальное оборудование для научных работ

Для данной работы не потребовалось никакого специального оборудования, вся разработка велась на уже имеющемся ноутбуке с установленными программными продуктами (куплен по цене 60 000 руб.).

Рассчитаем амортизацию оборудования. Срок полезного использования ПК составляет 3 года, согласно сроку полезного использования офисных машин.

Норма амортизации рассчитывается по формуле:

$$A_n = \frac{1}{n} * 100\% \quad (7)$$

где A_n – норма амортизации в процентах;

n – срок полезного использования в месяцах.

Подставив срок полезного использования ПК в формулу 7, получим:

$$A_n = \frac{1}{36} * 100\% = 2,78\%$$

Ежемесячные амортизационные отчисления находят по формуле:

$$A_m = \frac{C * A_n}{100\%} \quad (8)$$

Где A_m – ежемесячные амортизационные отчисления в рублях;

C – первоначальная стоимость оборудования в рублях;

A_n – норма амортизации в процентах.

Итоговая сумма амортизационных отчислений определяется по формуле:

$$A = A_m * n \quad (9)$$

где A – итоговая сумма амортизационных отчислений в рублях;

A_m – ежемесячные амортизационные отчисления в рублях;

n – срок использования ПК для исследования в месяцах.

Срок использования ПК для написания ВКР при первом исполнении – 4,6 месяцев, при втором – 3,1.

Таким образом, найдем итоговое значение амортизационных отчислений.

Для первого исполнения:

$$A = \frac{C * A_n}{100\%} * n = 60\,000 * 4,6 * 0,0278 = 7\,672,8 \text{ рублей}$$

Для второго исполнения:

$$A = \frac{C * A_n}{100\%} * n = 60\,000 * 3,1 * 0,0278 = 5\,170,8 \text{ рублей}$$

Исследование выполнялось с использованием программного обеспечения с бесплатной студенческой лицензией, поэтому амортизация на ПО не рассчитывается, а значит, итоговая сумма затрат на специальное оборудование равна сумме амортизации ПК, то есть 7 672,8 (5 170,8) рублей.

5.3.4.3 Основная заработная плата исполнителей темы

Данная статья затрат включает основную заработную плату, премии и доплаты всех исполнителей проекта. В качестве исполнителей проекта

выступают студенты и научный руководитель. Заработная плата рассчитывается по формуле:

$$З_{зп} = З_{осн} + З_{доп}, \quad (10)$$

где $З_{осн}$ – основная заработная плата;

$З_{доп}$ – дополнительная заработная плата.

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн} = \frac{З_{м} \cdot М}{F_{д}}, \quad (11)$$

где $З_{м}$ – месячный должностной оклад работника, руб.;

$М$ – количество месяцев работы без отпуска в течение года, равный 10,4 для 6 дневной рабочей недели и отпуске продолжительностью 48 дней;

$F_{д}$ – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн.

Таблица 8 – баланс рабочего времени

Показатели рабочего времени	Научный руководитель	Инженер программист.
Календарное число дней	365	365
Количество нерабочих дней	66	66
Потери рабочего времени -отпуск и невыходы по болезни	55	71
Действительный годовой фонд рабочего времени	244	228

Месячный оклад работника рассчитывается по формуле:

$$З_{м} = З_{тс} \cdot k_{общ}, \quad (12)$$

где $З_{тс}$ – заработная плата по тарифной ставке, руб.;

$k_{общ}$ – общий коэффициент, находится по формуле 13.

$$k_{общ} = (1 + k_{пр} + k_{д}) \cdot k_{р}, \quad (13)$$

$k_{пр}$ – премиальный коэффициент, равный 0,3;

$k_{д}$ – коэффициент доплат и надбавок, равный 0,15;

$k_{р}$ – районный коэффициент, равный 1,3 для Томска.

$$k_{общ} = (1 + 0,3 + 0,15) * 1,3 = 1,885$$

Таблица 9 – Затраты на заработную плату

Должность	$З_{тс}$ руб./мес.	$З_{дн}$, руб./день	T_p , раб. дни		$k_{общ}$	$З_{осн}$, руб.	
			исп1	исп2		исп1	исп2
Научный руководитель	34 187	1 457, 15	7	7	1,885	19 227, 09	19 227, 09
Инженер	14 512	661,95	138	93	1,885	172 193, 05	116 043, 15
Итого:						191 420,14	135 270,24

5.3.4.4 Дополнительная заработная плата исполнительной системы

Расчет дополнительной заработной платы ведется по следующей формуле:

$$З_{доп} = З_{осн} \cdot k_{доб}, \quad (14)$$

где $З_{доп}$ – дополнительная заработная плата, рубли;

$k_{доб}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12–0,15);

$З_{осн}$ – основная заработная плата, рубли.

Расчеты затрат на дополнительную заработную плату и заработную плату приведены ниже.

Для первого исполнения:

$$З_{\text{доп}} = 191\,430,14 \cdot 0,12 = 22\,971,62 \text{ руб}$$

$$З_{\text{зп}} = 191\,430,14 + 22\,971,62 = 214\,401,76 \text{ руб}$$

Для второго исполнения:

$$З_{\text{доп}} = 135\,270,24 \cdot 0,12 = 16\,232,43 \text{ руб}$$

$$З_{\text{зп}} = 135\,270,24 + 16\,232,43 = 151\,502,67 \text{ руб}$$

5.3.4.5 Отчисления во внебюджетные фонды (страховые отчисления)

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования, пенсионного фонда и медицинского страхования от затрат на оплату труда работников.

Отчисления во внебюджетные фонды рассчитывается по формуле:

$$З_{\text{внеб}} = З_{\text{зп}} \cdot k_{\text{внеб}}, \quad (15)$$

где $k_{\text{внеб}}$ – коэффициент отчислений во внебюджетные фонды.

Затраты на единый социальный налог составляют 30 % от полной заработной платы по проекту.

Основываясь на формуле 15, рассчитаем затраты на внебюджетный фонды для первого и второго исполнения.

Для первого исполнения:

$$З_{\text{внеб}} = 214\,401,76 \cdot 0,3 = 64\,320,53 \text{ руб.}$$

Для второго исполнения:

$$З_{\text{внеб}} = 151\,502,67 \cdot 0,3 = 45\,450,80 \text{ руб.}$$

5.3.4.6 Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать, оплата услуг связи, электроэнергия и т.д.

Накладные расходы вычисляются по формуле:

$$З_{\text{нкл}} = (\text{сумма статей}) \cdot k_{\text{нр}}, \quad (15)$$

где $k_{\text{нр}}$ – коэффициент накладных расходов, равен 16%.

Расчеты накладных расходов для разных исполнений сведены в таблицу 10.

Таблица 10 – Накладные расходы

Статьи затрат	Сумма, руб.	
	Исп1.	Исп2.
Затраты на специальное оборудование	7 672,8	5 170,8
Заработная плата (основная и дополнительная)	214 401,76	151 502,67
Отчисления во внебюджетные фонды	64 320,53	45 450,80
Сумма затрат	286 395,09	202 124,27
Коэффициент	0,16	0,16
Накладные расходы	45 823,21	32 339,88

5.3.4.7 Формирование бюджета затрат научно-исследовательского проекта

Бюджет затрат НТИ представляет собой сумму всех ранее рассмотренных статей. Тогда получим:

Для первого исполнения бюджет равен 332 218,3 руб., для второго – 224 464,15 руб.

5.4 Определение ресурсной, финансовой, бюджетной, социальной и экономической эффективности исследования

Интегральный финансовый показатель разработки определяется как:

$$I_{\text{фин } p}^{\text{исп } i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \quad (16)$$

где $I_{\text{фин } p}^{\text{исп } i}$ – интегральный финансовый показатель разработки;

Φ_{pi} – стоимость i-го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно- исследовательского проекта (в т.ч. аналоги).

Интегральный показатель ресурсоэффективности может быть вычислен по следующей формуле:

$$I_{pi} = \sum a_i * b_i , \quad (17)$$

где I_{pi} – интегральный показатель ресурсоэффективности для i-го варианта исполнения разработки;

a_i – весовой коэффициент i-го варианта исполнения разработки;

b_i – оценка i-го варианта исполнения разработки, выраженная в баллах, устанавливается экспертным путем по выбранной шкале оценивания.

Таблица 11 – Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Весовой коэффициент параметра	Исп.1	Исп.2
Способствует росту производительности труда пользователя	0,10	5	5
Сроки выполнения	0,15	3	5
Функциональная мощность	0,25	5	5
Удобство в эксплуатации	0,25	5	5
Потребность в ресурсах памяти	0,10	4	5
Надежность	0,15	4	5
Итого	1	4,45	5

$$I_{pi-исп1} = 5 \cdot 0,1 + 3 \cdot 0,15 + 5 \cdot 0,25 + 5 \cdot 0,25 + 4 \cdot 0,1 + 4 \cdot 0,15 = 4,45$$

$$I_{pi-исп2} = 5 \cdot 0,1 + 5 \cdot 0,15 + 5 \cdot 0,25 + 5 \cdot 0,25 + 5 \cdot 0,1 + 5 \cdot 0,15 = 5$$

Интегральный показатель эффективности вариантом использования определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{\text{исп } i} = \frac{I_{\text{р-исп } i}}{I_{\text{фин } p}} \quad (18)$$

Сравнительная эффективность проекта:

$$\mathcal{E}_{\text{ср}} = \frac{I_{\text{исп } 1}}{I_{\text{исп } 2}} \quad (19)$$

Используя данные формулы, можно найти наиболее эффективный вариант исполнения, результаты представлены в таблице 12.

Таблица 12 – Сравнительная эффективность разработки

№	Показатели	Исп.1	Исп.2
1	Интегральный финансовый показатель разработки	1	0,676
2	Интегральный показатель ресурсоэффективности разработки	4,45	5
3	Интегральный показатель эффективности	4,45	7,4
4	Сравнительная эффективность вариантов исполнения	0,6	1

По проведенным расчетам можно сделать вывод, что второй вариант исполнения является более экономически эффективным, так как его сравнительная эффективность больше, чем у первого исполнения.

Вывод по разделу

В рамках данного раздела был проведен комплексный SWOT-анализ проекта, который позволил выявить его сильные и слабые стороны. Также была произведена оценка качества и перспективности данного проекта, его конкурентоспособность. Разработка имеет мощный функционал, за счет чего прямые конкуренты отсутствуют.

Еще одним ключевым моментом, рассмотренным в данном разделе, является планирование работ, выполняемых в рамках проекта. Составленный список необходимых работ с распределением исполнителей, а также

вычисленная трудоемкость, позволили построить наглядный план-график работ в виде диаграммы Ганта. Разработка займет 93 дня.

Кроме того, определен бюджет проекта, затраты на заработные платы исполнителям с учетом страховых отчислений, рассчитаны накладные расходы, а также амортизация оборудования, задействованного в процессе выполнения проекта. Потенциальная стоимость проекта составляет 224 464,15 руб.

Также сравнительная оценка характеристик показала, что нужно отдать предпочтение второму варианту исполнения, за счет сроков и надежности разработки, которые основываются на знаниях инженера.

6 Социальная ответственность

Проект, сделанный в рамках исследовательской работы посвящен разработке мобильного приложения под операционную систему Android для скилл-шеринг сообщества «IT Way». Приложение позволит объединить весь разнородный контент, (статьи, новости, видео и подкасты), производимый сообществом, в одном месте для комфортного получения образовательной информации пользователем, что и отражает социальную направленность проекта.

Рабочая зона располагается в городе Томске. В качестве рабочего места рассматривается рабочее место программиста, для полнофункционального управления разработкой на котором находится ноутбук, мышь, а также для оперативного отслеживания процесса выполнения различных задач проекта имеется смартфон, подключаемый к ноутбуку по проводу. Использование электронно-вычислительной машины и сам процесс разработки влекут за собой ряд вредных и опасных факторов, которые в свою очередь сказываются на производительности труда и здоровье.

В данном разделе рассмотрены правовые и организационные вопросы обеспечения безопасности, проанализированы вредные и опасные факторы производственной среды программиста, разработаны действия, которые способны снизить воздействие данных факторов, кроме того освещены факторы, влияющие на окружающую среду, приведены меры безопасности при возникновении чрезвычайных ситуаций на рабочем месте.

6.1 Правовые и организационные вопросы обеспечения безопасности

6.1.1 Специальные правовые нормы трудового законодательства

Целями трудового законодательства являются установление государственных гарантий трудовых прав и свобод граждан, создание благоприятных условий труда, защита прав и интересов работников и работодателей [10].

Вид трудовой деятельности на ПК в рамках данной работы относится к группе В – творческая работа в режиме диалога с компьютером, не более 6 часов непосредственной работы за ПК.

Согласно этому, при 8-часовой рабочей смене перерывы следует устанавливать через 1,5 -2,0 часа от начала рабочей смены и через 1,5-2,0 часа после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы [11]. В течении смены также должен быть предоставлен перерыв для питания продолжительностью не менее 30 минут и не более двух часов. Продолжительность рабочего дня должна быть не меньше указанного времени в договоре, но не больше 40 часов в неделю. Всем работникам предоставляются выходные дни.

Организация-работодатель выплачивает заработную плату работникам. Возможно удержание заработной платы только в случаях, установленных ТК РФ ст. 137. В случае задержки заработной платы более чем на 15 дней, работник имеет право приостановить работу, письменно уведомив работодателя. Сверхурочная работа оплачивается за первые два часа работы не менее чем в полуторном размере, за последующие часы – не менее чем в двойном размере. По желанию работника сверхурочная работа может компенсироваться предоставлением дополнительного времени отдыха, но не менее времени, отработанного сверхурочно. Работодатель обязан предоставить отпуск без сохранения заработной платы работникам, обучающимся по имеющим государственную аккредитацию программам бакалавриата по очной форме обучения, совмещающим получение образования с работой, для прохождения промежуточной аттестации – 15 календарных дней в учебном году, для подготовки и защиты выпускной квалификационной работы и сдачи итоговых государственных экзаменов – четыре месяца, для сдачи итоговых государственных экзаменов – один месяц. Законодательством РФ запрещена дискриминация по любым признакам и принудительный труд [10].

6.1.2 Основные требования к компоновке рабочей зоны

Основными элементами рабочего места разработчика являются: рабочий стол, рабочий стул (кресло), дисплей, клавиатура, мышь (можно заменить одним ноутбуком); вспомогательными – пюпитр, подставка для ног [12].

Требования к рабочему столу, согласно ГОСТ Р 50923-96 [12]:

- Регулируемая высота стола должна изменяться в пределах от 680 до 800 мм, нерегулируемая должна составлять 725 мм.
- Предпочтительные размеры рабочей поверхности стола – глубина 800 мм, ширина 1600 мм.
- Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

Требования к рабочему стулу определяются по ГОСТ 21889-76 [13]:

- Кресло должно обеспечивать рациональную рабочую позу.
- Кресло должно обеспечивать условия для отдыха в кресле.
- Кресло должно создавать условия поддержания корпуса с сохранением естественных изгибов позвоночника.
- Кресло оператора должно включать следующие основные элементы: сиденье, спинку и подлокотники. В конструкцию кресла могут быть включены также дополнительные элементы – подголовник и подставка для ног.
- В конструкции кресла должны регулироваться высота поверхности сиденья и угол наклона спинки.

Взаимное расположение элементов рабочего места должно способствовать оптимальному режиму труда и отдыха, снижению утомления оператора, предупреждению появления ошибочных действий [14].

Дисплей на рабочем месте должен быть установлен ниже уровня глаз оператора. Угол наблюдения экрана оператором относительно горизонтальной линии взгляда не должен превышать 60° [12]. Часто используемые средства отображения информации, требующие менее точного и быстрого считывания показаний, допускается располагать в вертикальной плоскости под углом $\pm 30^\circ$

от нормальной линии взгляда и редко используемые средства отображения информации допускается располагать в горизонтальной плоскости под углом $\pm 60^\circ$ от сагиттальной плоскости [15].

6.2 Производственная безопасность

Для обеспечения производственной безопасности необходимо оценить воздействия на человека вредных (могут привести к заболеванию) и опасных (могут привести к травме) производственных факторов, которые могут возникать при разработке или эксплуатации проекта. В таблице 13 представлены факторы, характерные для работы, выполняемых на ПЭВМ.

Таблица 13 – Возможные вредные и опасные производственные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ		Нормативные документы
	Разработка	Эксплуатация	
1. Отсутствие или недостаток необходимого естественного освещения;	+	+	СП 52.13330.2016 [16]
2. Умственное перенапряжение;	+	-	ГОСТ Р ИСО 10075-1-2019 [17-19]
3. Монотонность труда;	+	-	СанПиН 1.2.3685-21 [20]
4. Физическая перегрузка, связанная с работой сидя;	+	-	Р 2.2.2006-05 [21]
5. Опасные факторы, связанные с электрическим током	+	-	ГОСТ 12.1.009-2017 [22]

6.2.1 Анализ опасных и вредных производственных факторов

6.2.1.1 Отсутствие или недостаток необходимого естественного освещения

Помещение с постоянным пребыванием людей должны иметь естественное освещение – освещение помещений светом неба, проникающим

через световые проемы в наружных ограждающих конструкциях. Помещение с недостаточным естественным светом – помещение, в котором коэффициент естественной освещенности ниже нормируемого.

Отсутствие природного света ухудшает самочувствие, работоспособность, способствует усталости глаз. Поэтому необходимо создать систему освещения рабочего места, отвечающую нормам. При проектировании естественного освещения определяются расчетные значения коэффициента естественного освещения. КЕО должен быть не меньше 1,2% для помещений, где выполняются работы с зрительным разрядом А.

По разным причинам природный свет не может быть стабильным, соответственно, для данной работы нужно использовать смешанный тип освещения. Освещенность на поверхности рабочего стола в зоне размещения документа должна быть 300 – 500 лк, что может достигаться установкой местного освещения, не создающего бликов на поверхности экрана. В случае, если естественного света много, следует использовать светозащитные устройства для окон.

6.2.1.2 Умственное перенапряжение

Информационная нагрузка, связанная с планированием, выполнением, контролем и оценкой задач, влияет на разработчика, тем самым вызывая умственное напряжение. Которое в свою очередь может иметь положительные последствия (активизация, обучение, повышение квалификации, опыт) и отрицательные (умственная усталость, стресс, эмоциональное выгорание) [17].

Умственная нагрузка является сложным и многомерным понятием, поэтому для оценки измерений умственной нагрузки не может быть применена единственная процедура. Не существует наилучшего способа измерений умственной нагрузки, так как наиболее подходящий способ измерений умственной нагрузки зависит от цели оценки [18].

Для того чтобы избежать отрицательного влияния рабочей системы на пользователей, необходимо подстраивать ее под пользователя. Проектируемая

рабочая система с самого начала требует учета человеческих, технологических и организационных факторов и их взаимодействий. Помимо количественных аспектов должны быть учтены качественные различия в умственной нагрузке, например, различия между задачами, связанными с восприятием, и задачами, связанными с интенсивной нагрузкой на память. Для уменьшения усталости оператора при проектировании рабочих систем должна быть снижена и/или оптимизирована интенсивность нагрузки, ограничена продолжительность вовлеченности оператора в работу или изменено распределение работы, например, путем введения перерывов для отдыха [19].

6.2.1.3 Монотонность труда

Процесс разработки программы является монотонным трудом – однообразие трудовых операций – и вызывает монотонию – состояние организма человека, определяемое комплексом психологических и физических изменений. Например, замедление способности к переключению внимания, сонливость, развитие невротических и сосудистых расстройств. Допустимое число элементов или повторяющихся операций от 9 до 6 [20]. Для уменьшения отрицательных последствий монотонного труда следует: оптимизировать темп и ритм труда, автоматизировать однообразный ручной труд, чередовать задачи, прерываться от работы на отдых.

6.2.1.4 Физическая перегрузка, связанная с работой сидя

Физическая перегрузка, вызванная работой в сидячем положении, провоцирует бессонницу, венозное расширение, сбои в работе пищеварения. Согласно Р 2.2.2006-05 рекомендуется занять свободную, удобную позу с возможностью смены рабочего положения тела (сидя, стоя). Нахождение в позе стоя должно составлять до 40% времени смены [21].

6.2.1.5 Опасные факторы, связанные с электрическим током

Поражение электрическим током является опасным производственным фактором и, поскольку оператор ПЭВМ имеет дело с электрооборудованием, то вопросам электробезопасности на его рабочем месте должно уделяться много внимания.

Электробезопасность – система организационных и технических мероприятий и средств, обеспечивающих защиту персонала от вредного и опасного воздействия электрического тока, электрической дуги, электромагнитного поля и статического электричества.

Для оператора ПЭВМ при работе с электрическим оборудованием обязательны следующие меры предосторожности:

- Перед началом работы нужно убедиться, что выключатели и розетка закреплены и не имеют оголённых токоведущих частей;
- При обнаружении неисправности оборудования и приборов необходимо сообщить человеку, ответственному за оборудование.

6.2.2 Обоснование мероприятий по снижению уровней воздействия опасных и вредных факторов на работающего

Для снижения уровня воздействия вредных и опасных факторов на работника необходимо грамотно спланировать размещение сотрудников в помещении, а также их работу, обеспечить перерывы от работы, поддерживать оптимальным освещение, ознакомить сотрудников с правилами безопасности.

6.3 Экологическая безопасность

Мобильное приложение, разрабатываемое в рамках работы, не наносит вреда окружающей среде. Однако, для самой разработки используются средства, необходимые для нее – это ноутбук и телефон, при поломке которые подлежат специальной утилизации для 4 класса опасности. Помимо этого, на рабочем столе располагается лампа, канцелярия, которые утилизируются в соответствии со своим классом опасности. Также в процессе разработки, программист создает различный бытовой мусор, который тоже стоит утилизировать. Процедура

утилизации должна соответствовать ГОСТ 53692-2009 Обращение с отходами [23]. В противном случае, тяжелые металлы, кислоты и щелочи, содержащиеся в аккумуляторах, ртуть в лампах, могут поразить гидросферу и литосферу, отравить живых существ. Пищевой мусор в большинстве своем сжигается, тем самым в атмосферу выбрасываются тяжелые металлы и диоксины, которые вызывают рак, аллергию, бесплодие.

Потребление тепло и электроэнергии необходимо для отопления, приготовления пищи, зарядки и работы различной техники. Производство энергии существенно влияет на экологическую ситуацию. Сжигание твердого и жидкого топлива сопровождается выделением газов, оксидов, пыли, сажи и других загрязняющих атмосферу веществ. Строительство и эксплуатация ГЭС приводят к потере леса и плодородных земель, к уничтожению некоторых видов рыб, увеличивают риск возникновения землетрясений. Рациональным решением можно считать энергосбережение, а также мероприятия, направленные на переход к безотходным и малоотходным технологиям.

6.4 Безопасность в чрезвычайных ситуациях

В рабочей среде оператора ПЭВМ в современных условиях возможно возникновение следующих чрезвычайных ситуаций:

- Компьютерный терроризм (социальная ЧС);
- Угроза пандемии (биологическая и социальная ЧС);
- Пожар (техногенная ЧС).

Наиболее характерной для рабочего помещения, снабженного электронно-вычислительными машинами, чрезвычайной ситуацией является пожар.

Офисное помещение, согласно СП 12.13130.2009, имеет категорию В пожароопасности, так как в помещение обращаются горючие и трудногорючие вещества и материалы, которые горят без образования взрывоопасных смесей и только при контакте с кислородом, водой или друг с другом [24].

Согласно ФЗ от 22.07.2008 №123 составлена таблица 14 [25].

Таблица 14 – Действия предотвращения пожара и действия в случае возникновения пожара

Меры по предотвращению пожара:	В случае возникновения пожара необходимо:
Периодическое проведение инструктажей по пожарной безопасности.	Немедленно сообщить о пожаре в ближайшую пожарную часть, администрации организации.
Периодическое проведение учебных тревог.	Выключить приточно-вытяжную вентиляцию.
Наличие плана-эвакуации для каждого из помещений.	Немедленно оповестить персонал о пожаре с помощью установленной системы оповещения.
Наличие огнетушителей или других средств для тушения пожара.	Открыть все эвакуационные выходы из здания.
Наличие в помещении пожарной сигнализации.	Быстро, без паники и суеты эвакуироваться из здания согласно плану эвакуации, избегая встречных и пересекающих потоков людей.
Соблюдение правил пожарной безопасности при установке и эксплуатации различных электроприборов в помещении.	Покидая помещение, отключить все электроприборы, выключить свет, плотно закрыть за собой двери, окна и форточки во избежание распространения огня и дыма в смежные помещения.
Периодические проверки пожарной инспекцией согласно графику.	Администрации организовать встречу работников пожарной охраны и проводить их к месту пожара.
	До приезда работников пожарной охраны членам добровольной пожарной дружины организовать тушение пожара первичными средствами

Вывод по разделу

Для рационального использования трудового времени необходимо организованно подойти к реализации условий труда сотрудника, провести мероприятия для повышения труда. Важно избавиться от постоянного воздействия вредных и опасных факторов и свести к минимуму вероятность возникновения чрезвычайных ситуаций, а также обеспечить безопасность персонала в случае их возникновения. Кроме того, нужно заботиться об

экологичности, не допускать загрязнения окружающей среды, потому как ее загрязнение может привести к катастрофическим последствиям.

Заключение

В ходе выпускной квалификационной работы был реализован весь основной функционал мобильного образовательного приложения для комфортного потребления различного контента (подкасты, новости, видео) пользователями сообщества «IT Way».

Для успешной реализации проекта была изучена техническая база создания мобильного приложения, выбраны язык, среда разработки, а также плагины, которые способны помочь в создании аудиоплеера, видеоплеера, регистрации, хранения данных пользователя. Также был создан отдельный шрифт, предоставляющий иконки, проработан дизайн приложения: стили шрифтов и цвета.

Список источников

- 1 Digital 2021: the latest insights into the «state of digital» [Электронный ресурс]. – URL: <https://wearesocial.com/blog/2021/01/digital-2021-the-latest-insights-into-the-state-of-digital> (дата обращения: 27.05.2021);
- 2 Global smartphone market Q1 2021 [Электронный ресурс]. – URL: <https://www.canalys.com/newsroom/canalys-worldwide-smartphone-market-Q1-2021> (дата обращения: 27.05.2021);
- 3 Flutter vs React-Native: детальное сравнение [Электронный ресурс]. – URL: <https://tproger.ru/translations/flutter-vs-react-native-comparison/> (дата обращения: 07.05.2021);
- 4 Как пользоваться Visual Studio Code [Электронный ресурс]. – URL: <https://timeweb.com/ru/community/articles/chto-takoe-visual-studio-code> (дата обращения: 08.05.2021);
- 5 Что такое Android Studio [Электронный ресурс]. – URL: <http://web.spt42.ru/index.php/chto-takoe-android-studio> (дата обращения: 08.05.2021);
- 6 Введение в Firebase: пишем простое социальное приложение на Swift [Электронный ресурс]. – URL: <https://habr.com/ru/post/277941/> (дата обращения: 13.05.2021);
- 7 Основы Flutter для начинающих (Часть II) [Электронный ресурс]. – URL: <https://habr.com/ru/post/277941/> (дата обращения: 14.05.2021);
- 8 Пакет bloc 7.0.0 [Электронный ресурс]. – URL: <https://pub.dev/packages/bloc> (дата обращения: 16.05.2021);
- 9 Mobile Operating System Market Share Russian Federation [Электронный ресурс]. – URL: <https://gs.statcounter.com/os-market-share/mobile/russian-federation> (дата обращения: 6.05.2021);
- 10 Трудовой кодекс Российской Федерации: текст изм. и доп. на 1 мая 2021 г. [Электронный ресурс]. – URL:

http://www.consultant.ru/document/cons_doc_LAW_34683/4a31b57741dfbf7b99756f5338ab052b3fb209cc/ (дата обращения: 16.05.2021);

- 11 Типовая инструкция по охране труда при работе на персональном компьютере. [Электронный ресурс]. – URL: <https://mosmetod.ru/metodicheskoe-prostranstvo/muzyka/kabinet/tipovaya-instruktsiya-po-okhrane-truda-pri-rabote-na-personalnom-kompyutere.html#:~:text=%D0%92%D0%B8%D0%B4%D1%8B%20%D1%82%D1%80%D1%83%D0%B4%D0%BE%D0%B2%D0%BE%D0%B9%20%D0%B4%D0%B5%D1%8F%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D0%BE%D1%81%D1%82%D0%B8%20%D1%80%D0%B0%D0%B7%D0%B4%D0%B5%D0%BB%D1%8F%D1%8E%D1%82%D1%81%D1%8F%20%D0%BD%D0%B0,%D0%B2%20%D1%80%D0%B5%D0%B6%D0%B8%D0%BC%D0%B5%20%D0%B4%D0%B8%D0%B0%D0%BB%D0%BE%D0%B3%D0%B0%20%D1%81%20%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BE%D0%BC>. (дата обращения: 16.05.2021);
- 12 ГОСТ Р 50923-96 Дисплей. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200025975> (дата обращения: 16.05.2021);
- 13 ГОСТ 21889-76 Система «Человек-машина». Кресло человека-оператора. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200012832> (дата обращения: 16.05.2021);
- 14 ГОСТ 22269-76 Система «Человек-машина». Рабочее место оператора. Взаимное расположение элементов рабочего места. Общие эргономические требования и требования к

- производственной среде. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200012834> (дата обращения: 16.05.2021);
- 15 ГОСТ 12.2.033-78 Система стандартов безопасности труда (ССБТ). Рабочее место при выполнении работ сидя. Общие эргономические требования. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200003913> (дата обращения: 16.05.2021);
- 16 СП 52. 13330.2016 Естественное и искусственное освещение. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/456054197> (дата обращения: 17.05.2021);
- 17 ГОСТ Р ИСО 10075-1-2019 Эргономические принципы обеспечения адекватности умственной нагрузки. Часть 1. Основные понятия, принципы и определения. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200167488> (дата обращения: 17.05.2021);
- 18 ГОСТ Р ИСО 10075-3-2009 Эргономические принципы обеспечения адекватности умственной нагрузки. Часть 3. Принципы и требования к методам измерений и оценке умственной нагрузки. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200075947> (дата обращения: 17.05.2021);
- 19 ГОСТ Р ИСО 10075-2-2009 Эргономические принципы обеспечения адекватности умственной нагрузки. Часть 2. Принципы проектирования. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200075944> (дата обращения: 17.05.2021);
- 20 СанПин 1.2.3685-21 Гигиенические нормативы и требования к обеспечению безопасности и безвредности для человека факторов

- среды обитания. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/573500115> (дата обращения: 18.05.2021);
- 21 Р 2.2.2006-05 Гигиена труда. Руководство по гигиенической оценке факторов рабочей среды и трудового процесса. Критерии и классификация условий труда. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200040973> (дата обращения: 18.05.2021);
- 22 ГОСТ 12.1.009-2017 Система стандартов безопасности труда. Электробезопасность. Термины и определения. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200161311> (дата обращения: 19.05.2021);
- 23 ГОСТ Р 53692-2009 Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200081740> (дата обращения: 20.05.2021);
- 24 СП 12.13130.2009 Определение категорий помещений, зданий и наружных установок по взрывопожарной и пожарной опасности. [Электронный ресурс]. – URL: <https://docs.cntd.ru/document/1200071156> (дата обращения: 21.05.2021);
- 25 Федеральный закон Технический регламент о требованиях пожарной безопасности от 22.07.2008 № 123-ФЗ. [Электронный ресурс]. – URL: http://www.consultant.ru/document/cons_doc_LAW_78699/ (дата обращения: 22.05.2021).

Приложение А

Листинг podcast_cubit.dart

```
import 'dart:async';
import 'dart:convert';
import 'package:http/http.dart' as http;
import 'package:bloc/bloc.dart';
import 'package:it_way_app/blocs/podcast/podcast_state.dart';
import 'package:it_way_app/data/podcast_data.dart';

class PodcastCubit extends Cubit<PodcastStates> {
  PodcastCubit() : super(null);

  Future<List<PodcastData>> getPodcasts() async {
    emit(PodcastStates.dataLoading());

    final json = await getjson();
    final List<PodcastData> podcasts = [];

    for (final jsonPodcast in json) {
      final attributes = jsonPodcast['attributes'];
      final int podcastNumber = attributes['number'];
      final String podcastTitle = attributes['title'];
      final String podcastImageURL = attributes['image'];
      final String publishedAt = attributes['published-at'];
      final String playingTimeOfPodcast = attributes['duration'];
      final String podcastFile = attributes['file-url'];

      PodcastData podcast = PodcastData(
        playingTimeOfPodcast: playingTimeOfPodcast,
        podcastImageURL: podcastImageURL,
        podcastNumber: podcastNumber,
        podcastTitle: podcastTitle,
        publishedAt: publishedAt,
        audioFile: podcastFile,
      );
      podcasts.add(podcast);
    }

    emit(PodcastStates.dataLoaded(podcasts));
  }

  dynamic getjson() async {
    final response = await http.get(
      Uri.parse('http://red-magic.ru/api/v1/records?model=Podcast::Episode'));
    final json = jsonDecode(response.body);
    return json['data'];
  }
}
```

Листинг podcast_state.dart

```
import 'package:it_way_app/data/podcast_data.dart';

class PodcastStates {
  const PodcastStates._(
    {this.status = PodcastStatus.dataLoading, this.podcats});

  const PodcastStates.dataLoaded(List<PodcastData> podcats)
    : this._(
        status: PodcastStatus.dataLoaded,
        podcats: podcats,
      );

  const PodcastStates.failure() : this._(status: PodcastStatus.failure);

  const PodcastStates.dataLoading() : this._(status: PodcastStatus.dataLoading);

  final PodcastStatus status;
  final List<PodcastData> podcats;
}

enum PodcastStatus { dataLoaded, failure, dataLoading }
```

Листинг podcast_card.dart

```
import 'package:flutter/material.dart';
import 'package:it_way_app/data/podcast_data.dart';

import 'package:it_way_app/statics/style.dart';
import 'package:it_way_app/screen_navigation.dart';
import 'package:it_way_app/statics/navigation.dart';

class PodcastCard extends StatefulWidget {
  const PodcastCard({
    @required this.data,
  });

  final PodcastData data;
  @override
  _PodcastCardState createState() => _PodcastCardState();
}

class _PodcastCardState extends State<PodcastCard> {
  String get podcastTitle => widget.data.podcastTitle;
  String get podcastNumber => widget.data.podcastNumber.toString();
  String get podcastImage => widget.data.podcastImageURL;
  String get playingTimeOfPodcast => widget.data.playingTimeOfPodcast;
  String get publishedAt => widget.data.publishedAt;

  @override
```

```

Widget build(BuildContext context) {
  return Container(
    padding: EdgeInsets.fromLTRB(10, 15, 10, 0),
    height: MediaQuery.of(context).size.width,
    child: Card(
      shape: RoundedRectangleBorder(
        borderRadius: BorderRadius.circular(15.0),
      ),
      elevation: 5,
      child: InkWell(
        onTap: () {
          ScreenNavigation.routeTo(
            route: LibraryRoutes.onePodcastScreen,
            context: context,
            bundle: <String, dynamic>{
              'data': widget.data,
            },
          );
        },
        child: Stack(
          children: [
            Container(
              decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(15.0),
                image: DecorationImage(
                  image: podcastImage != null
                    ? NetworkImage(podcastImage)
                    : AssetImage('assets/images/podcast.jpg'),
                  fit: BoxFit.cover,
                ),
            ),
            Container(
              decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(15.0),
                color: Colors.black.withOpacity(0.4),
            ),
            Container(
              decoration: BoxDecoration(
                borderRadius: BorderRadius.circular(15.0),
            ),
            child: Padding(
              padding: const EdgeInsets.all(15.0),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.spaceBetween,
                crossAxisAlignment: CrossAxisAlignment.start,
                children: <Widget>[
                  Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [

```

```

        Row(
          children: [
            Icon(
              Icons.timer,
              color: Colors.white,
              size: 12.0,
            ),
            SizedBox(width: 5.0),
            Text(
              playingTimeOfPodcast,
              style: LibraryStyle.typeContentStyle,
            ),
            SizedBox(width: 5.0),
            Text(
              publishedAt,
              style: LibraryStyle.typeContentStyle,
            ),
          ],
        ),
        Text(
          podcastTitle,
          style: LibraryStyle.headingPodcastAndNews,
          textAlign: TextAlign.justify,
        ),
      ],
    ),
    SizedBox(height: 5.0),
    Text("Эпизод " + podcastNumber,
      style: LibraryStyle.subtitleStyle),
  ],
),
),
),
],
),
),
);
}
}

```

Листинг podcast_screen.dart

```

import 'package:flutter/material.dart';
import 'package:it_way_app/blocs/podcast/podcast_cubit.dart';
import 'package:it_way_app/blocs/podcast/podcast_state.dart';
import 'package:it_way_app/components/loading.dart';
import 'package:it_way_app/components/podcast_card.dart';
import 'package:it_way_app/components/appBar.dart';

```



```

import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:it_way_app/data/podcast_data.dart';
import 'package:it_way_app/statics/colors.dart';

class PodcastScreen extends StatefulWidget {
  @override
  _PodcastScreen createState() => _PodcastScreen();
}

class _PodcastScreen extends State<PodcastScreen> {
  PodcastCubit podcastScreenCubit = PodcastCubit();

  @override
  void initState() {
    podcastScreenCubit.getPodcasts();
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBarWidget(typeOfContent: 'Подкасты'),
      body: SafeArea(
        child: BlocBuilder<PodcastCubit, PodcastStates>(
          bloc: podcastScreenCubit,
          builder: (BuildContext context, PodcastStates state) {
            if (state.status == PodcastStatus.dataLoading) {
              return LoadingApp();
            }
            if (state.status == PodcastStatus.failure) {}
            if (state.status == PodcastStatus.dataLoaded) {
              final List<PodcastData> listPodcatData = state.podcats;
              return SingleChildScrollView(
                physics: BouncingScrollPhysics(),
                child: Column(
                  children: buildPodcasts(listPodcatData),
                ),
              );
            }
            return Container();
          },
        ),
      ),
    );
  }

  List<PodcastCard> buildPodcasts(List<PodcastData> listPodcatData) {
    List<PodcastCard> listPodcasts = [];
    for (final podcastData in listPodcatData) {
      listPodcasts.add(

```

```

        PodcastCard(
          data: podcastData,
        ),
      );
    }
    return listPodcasts;
  }
}

```

Листинг one_podcast_screen.dart

```

import 'dart:async';

import 'package:flutter/material.dart';
import 'package:audio_service/audio_service.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:it_way_app/audio_player/error_bar.dart';
import 'package:it_way_app/audio_player/features/audio_track/domain/entities/audio_track.dart';
import 'package:it_way_app/audio_player/features/show_error/error_bar_cubit.dart';
import 'package:it_way_app/audio_player/music_player/music_player.dart';
import 'package:it_way_app/data/podcast_data.dart';

import 'package:it_way_app/audio_player/injection_container.dart';
import 'package:flutter/services.dart';

class OnePodcastScreen extends StatefulWidget {
  const OnePodcastScreen({
    @required this.data,
  });
  final PodcastData data;

  @override
  _OnePodcastScreenState createState() => _OnePodcastScreenState();
}

class _OnePodcastScreenState extends State<OnePodcastScreen> {
  String get podcastImage => widget.data.podcastImageURL;
  String get podcastTitle => widget.data.podcastTitle;
  String get podcastNumber => widget.data.podcastNumber.toString();
  static const _errorBarHeight = 80.0;
  AudioTrack audioTrack;

  @override
  void initState() {
    audioTrack = AudioTrack(
      widget.data.audioFile,
      widget.data.podcastNumber.toString(), // тут должно быть описание
      widget.data.podcastImageURL,
      widget.data.podcastTitle,
    );
  }
}

```

```

);
super.initState();
}

@override
Widget build(BuildContext context) {
  SystemChrome.setSystemUIOverlayStyle(
    SystemUiOverlayStyle(statusBarBrightness: Brightness.dark));

  return AudioServiceWidget(
    child: Scaffold(
      body: MultiBlocProvider(
        providers: [
          BlocProvider.value(value: serviceLocator<ErrorBarCubit>())
        ],
        child: BlocListener<ErrorBarCubit, ErrorBarStateAbstract>(
          listener: (context, state) {
            if (state is ShowErrorBarState) {
              Timer(Duration(seconds: 5), () {
                serviceLocator<ErrorBarCubit>().hide();
              });
            }
          }, child: BlocBuilder<ErrorBarCubit, ErrorBarStateAbstract>(
            builder: (context, state) {
              final message = state is ShowErrorBarState ? state.message : '';
              final position =
                state is ShowErrorBarState ? 0.0 : _errorBarHeight * -1;
              return Stack(
                children: [
                  MusicPlayer(
                    track: audioTrack,
                  ),
                  AnimatedPositioned(
                    duration: Duration(milliseconds: 500),
                    width: MediaQuery.of(context).size.width,
                    curve: Curves.easeInOutQuart,
                    top: position,
                    child: GestureDetector(
                      onVerticalDragUpdate: (details) {
                        int sensitivity = 1;
                        if (details.delta.dy < -sensitivity) {
                          serviceLocator<ErrorBarCubit>().hide();
                        }
                      },
                      child: ErrorBar(_errorBarHeight, message),
                    ),
                  ),
                ],
              );
            })),
          ),
        ),
      ),
    ),
  ),

```

```

    ),
  );
}
}

```

Листинг music_player.dart

```

import 'package:it_way_app/audio_player/features/audio_track/domain/entities/audio_track.dart';
import 'package:it_way_app/audio_player/features/media_player/bloc/media_player_cubit.dart';
import 'package:it_way_app/audio_player/injection_container.dart';
import 'package:it_way_app/audio_player/music_player/bottom_bar.dart';
import 'package:it_way_app/audio_player/music_player/progress_bar.dart';
import 'package:it_way_app/audio_player/music_player/top_bar.dart';
import 'package:flutter/material.dart';
import 'package:flutter/widgets.dart';
import 'package:flutter_bloc/flutter_bloc.dart';
import 'package:it_way_app/statics/colors.dart';
import 'package:it_way_app/statics/style.dart';

class MusicPlayer extends StatelessWidget {
  const MusicPlayer({this.track});

  final AudioTrack track;
  Widget build(BuildContext context) {
    final mediaQuery = MediaQuery.of(context);
    bool started = false;

    return MultiBlocProvider(
      providers: [
        BlocProvider.value(value: serviceLocator<MediaPlayerCubit>()),
      ],
      child: BlocBuilder<MediaPlayerCubit, MediaPlayerStateAbstract>(
        builder: (context, mediaPlayerState) {
          if (!started) {
            BlocProvider.of<MediaPlayerCubit>(context).play(track);
            started = true;
          }

          return Container(
            color: Colors.white,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.start,
              crossAxisAlignment: CrossAxisAlignment.center,
              children: [
                Padding(
                  padding: EdgeInsets.only(
                    top: mediaQuery.padding.top + 8, left: 10, right: 10),

```

```

        child: TopBar(),
      ),
      Divider(thickness: 0.5, color: LibraryColors.kUnActiveColor),
      Expanded(
        child: Padding(
          padding: EdgeInsets.symmetric(horizontal: 30),
          child: Column(children: [
            Row(
              mainAxisAlignment: MainAxisAlignment.max,
              mainAxisAlignment: MainAxisAlignment.center,
              children: [
                Expanded(
                  child: Text(
                    'Эпизод ${mediaPlayerState.audioTrack?.number ?? ''} ',
                    style: LibraryStyle.onePodcastHeading,
                    textAlign: TextAlign.center,
                  ),
                )
              ],
            ),
            SizedBox(height: 10),
            Stack(
              children: [
                SizedBox(
                  width: 300,
                  height: 300,
                  child: mediaPlayerState.audioTrack?.imageUrl != null
                    ? Image.network(
                        mediaPlayerState.audioTrack?.imageUrl
                      ) : Image.asset('assets/images/podcast.jpg'),
                ),
                (() {
                  if (mediaPlayerState
                    is MediaPlayerLoadingTrackState) {
                    return SizedBox(
                      width: 200,
                      height: 200,
                      child: Stack(
                        alignment: Alignment.center,
                        children: [
                          Container(
                            color: Colors.black.withOpacity(0.7),
                          ),
                          CircularProgressIndicator(),
                        ],
                      ),
                    );
                  }
                })
              ],
            ),
          ],
        ),
      ),
    ),
  ),
  return SizedBox();
}

```

```

        }()),
    ],
),
PreferredSize(height: 20),
Row(
  mainAxisAlignment: MainAxisAlignment.max,
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Expanded(
      child: Text(
        mediaPlayerState.audioTrack?.title ?? '',
        textAlign: TextAlign.center,
        style: LibraryStyle.onePodcastHeading,
      ),
    ),
  ],
),
PreferredSize(
  height: 6,
),
Row(
  mainAxisAlignment: MainAxisAlignment.max,
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Expanded(
      child: Text(
        '«IT Way»',
        overflow: TextOverflow.ellipsis,
        textAlign: TextAlign.center,
        style:
          TextStyle(color: LibraryColors.kUnActiveColor),
      ),
    ),
  ],
),
ProgressBar(),
]),
),
),
BottomBar(track),
],
),
);
}),
);
}
}

```

Листинг media_player_cubit.dart

```
import 'dart:async';

import 'package:audio_service/audio_service.dart';
import 'package:it_way_app/audio_player/features/audio_track/domain/entities/audio_track.dart';
import 'package:it_way_app/audio_player/features/audio_track/domain/usecases/get_audio_track_usecase.dart';
import 'package:it_way_app/audio_player/features/media_player/audio_player_task.dart';
import 'package:it_way_app/audio_player/features/media_player/playing_duration_cubit/playing_duration_cubit.dart';
import 'package:it_way_app/audio_player/features/media_player/playing_position_cubit/playing_position_cubit.dart';
import 'package:it_way_app/audio_player/features/media_player/playing_route_cubit/playing_route_cubit.dart';
import 'package:it_way_app/audio_player/features/media_player/playing_speed_bloc/playing_speed_bloc.dart';
import 'package:it_way_app/audio_player/features/media_player/player_seek_cubit/player_seek_cubit.dart';
import 'package:it_way_app/audio_player/features/show_error/error_bar_cubit.dart';
import 'package:bloc/bloc.dart';
import 'package:meta/meta.dart';

part 'media_player_state.dart';

void _entryPoint() {
  AudioServiceBackground.run(() => AudioPlayerTask());
}

class MediaPlayerCubit extends Cubit<MediaPlayerStateAbstract> {
  final PlayingRouteCubit playRouteCubit;
  final PlayingSpeedBloc playingSpeedBloc;
  final PlayingDurationCubit playingDurationCubit;
  final PlayingPositionCubit playingPositionCubit;
  final PlayerSeekCubit playerSeekCubit;
  final GetAudioTrackUseCase getAudioTrackUseCase;
  final ErrorBarCubit errorBarCubit;

  AudioTrack _audioTrack;

  get isPlayingThroughEarpiece => playRouteCubit.isPlayingThroughEarpiece;

  factory MediaPlayerCubit(
    PlayingRouteCubit playRouteCubit,
    PlayingSpeedBloc playingSpeedBloc,
    PlayingDurationCubit playingDurationCubit,
    PlayingPositionCubit playingPositionCubit,
    PlayerSeekCubit playerSeekCubit,
    GetAudioTrackUseCase getAudioTrackUseCase,
```

```

    ErrorBarCubit errorBarCubit,
  ) {
    var mediaPlayer = MediaPlayerCubit._init(
      playRouteCubit,
      playingSpeedBloc,
      playingDurationCubit,
      playingPositionCubit,
      playerSeekCubit,
      getAudioTrackUseCase,
      errorBarCubit,
    );

    _startAudioService(mediaPlayer);

    return mediaPlayer;
  }

  MediaPlayerCubit._init(
    this.playRouteCubit,
    this.playingSpeedBloc,
    this.playingDurationCubit,
    this.playingPositionCubit,
    this.playerSeekCubit,
    this.getAudioTrackUseCase,
    this.errorBarCubit,
  ) : super(MediaPlayerInitialState(null));

  static Future<void> _startAudioService(
    MediaPlayerCubit mediaPlayer,
  ) async {
    AudioService.customEventStream.listen((event) {
      if (event[AudioPlayerTask.DURATION_SECONDS_EVENT] != null) {
        mediaPlayer.changDuration(
          Duration(seconds: event[AudioPlayerTask.DURATION_SECONDS_EVENT]));
      }
      if (event[AudioPlayerTask.POSITION_SECONDS_EVENT] != null) {
        mediaPlayer.setPosition(
          Duration(seconds: event[AudioPlayerTask.POSITION_SECONDS_EVENT]));
      }

      if (event[AudioPlayerTask.PLAY_EVENT] != null) {
        mediaPlayer.resume();
      }
      if (event[AudioPlayerTask.PAUSE_EVENT] != null) {
        mediaPlayer.pause();
      }

      if (event[AudioPlayerTask.STOP_EVENT] != null) {
        mediaPlayer.stopFromIsolate();
      }
    });
  });

```



```

    await AudioService.start(backgroundTaskEntrypoint: _entryPoint)
      .then((value) =>
        print('=== successStartBackgroundTask ===>' + value.toString()))
      .onError((error, stackTrace) => print(
        '----- successStartBackgroundTask error -----: ' +
        error.toString()));
  }

void _addTrackAndPlay(AudioTrack track) {
  AudioService.updateQueue([]);
  AudioService.addQueueItem(MediaItem(
    id: track.url,
    album: '«IT Way»',
    title: track.title,
  ));

  AudioService.customAction(
    AudioPlayerTask.PLAY_ACTION, playingSpeedBloc.playingSpeed);
}

Future<dynamic> _addActionToAudioService(Function callback) async {
  if (!AudioService.running) {
    await _startAudioService(this);
  }

  return callback();
}

Future<void> play(AudioTrack track) async {
  if (AudioService.running) {
    await nextTrack(track);
  } else {
    emit(MediaPlayerLoadingTrackState(null));

    if (_audioTrack == null) {
      _audioTrack = await getAudioTrackUseCase.next(track: track);

      if (_audioTrack is AudioTrack) {
        _addActionToAudioService(() => _addTrackAndPlay(_audioTrack));
      } else {
        errorBarCubit.show('no tracks in library');
      }
    } else {
      _addActionToAudioService(
        () => AudioService.customAction(AudioPlayerTask.RESUME_ACTION));
    }
    emit(MediaPlayerPlayingState(_audioTrack));
  }
}

```

```

Future<void> pause() async {
  emit(MediaPlayerPausedState(_audioTrack));
  _addActionToAudioService(() => AudioService.pause());
}

Future<void> resume() async {
  emit(MediaPlayerPlayingState(_audioTrack));
  _addActionToAudioService(
    () => AudioService.customAction(AudioPlayerTask.RESUME_ACTION));
}

Future<void> nextTrack(AudioTrack track) async {
  if (_audioTrack is AudioTrack) {
    var nextTrack = await getAudioTrackUseCase.next(track: track);
    if (nextTrack is AudioTrack) {
      _audioTrack = nextTrack;
      emit(MediaPlayerPlayingState(_audioTrack));
      _addActionToAudioService(() => _addTrackAndPlay(nextTrack));
    } else {
      errorBarCubit.show('no more tracks');
    }
  }
}

Future<void> stop() async {
  emit(MediaPlayerStoppedState(_audioTrack));
  await _addActionToAudioService(
    () => AudioService.stop().then((value) => print('')));
}

Future<void> toggleEarpieceOrSpeakers() async {
  if (await _addActionToAudioService(() =>
    AudioService.customAction(AudioPlayerTask.EARPIECE_TOGGLE_ACTION))) {
    playRouteCubit.toggle();
  }
}

Future<void> increaseSpeed() async {
  playingSpeedBloc.add(PlayingSpeedIncreaseEvent(
    () {
      _addActionToAudioService(() => AudioService.customAction(
        AudioPlayerTask.SPEED_CHANGE_ACTION,
        playingSpeedBloc.playingSpeed));
    },
  ));
}

Future<void> seek(Duration newPosition) async {
  if (newPosition.inSeconds > 0) {
    newPosition = newPosition;
  }
}

```

```

    playerSeekCubit.seek(newPosition);
    _addActionToAudioService(() => AudioService.customAction(
        AudioPlayerTask.SEEK_ACTION, newPosition.inSeconds));
  }

  Future<void> stopFromIsolate() async {
    /// handle stop event from isolate and from UI to avoid send stop action
    /// after isolate already stopped
    _audioTrack = null;
    emit(MediaPlayerStoppedState(_audioTrack));
  }

  Future<void> changDuration(Duration duration) async {
    playingDurationCubit.setDuration(duration);
  }

  Future<void> setPosition(Duration position) async {
    playingPositionCubit.setPosition(position);
  }
}

```

Листинг one_video_screen.dart

```

import 'package:flutter/material.dart';

import 'package:flutter/cupertino.dart';
import 'package:it_way_app/components/appBarBack.dart';
import 'package:it_way_app/data/video_data.dart';
import 'package:it_way_app/statics/style.dart';
import 'package:youtube_player_flutter/youtube_player_flutter.dart';

import 'package:flutter/rendering.dart';
import 'package:flutter/services.dart';

class OneVideoScreen extends StatefulWidget {
  const OneVideoScreen({
    @required this.data,
  });
  final VideoData data;
  @override
  _OneVideoScreenState createState() => _OneVideoScreenState();
}

class _OneVideoScreenState extends State<OneVideoScreen> {
  String get videoPreview => widget.data.videoPreviewURL;
  String get videoDescription => widget.data.videoDescription;
  String get videoURL => widget.data.videoURL;
  String get videoTitle => widget.data.videoTitle;
  String videoId;
  YoutubePlayerController _controller;

```

```

@override
void initState() {
  videoId = YoutubePlayer.convertUrlToId(videoURL);
  _controller = YoutubePlayerController(
    initialVideoId: videoId,
    flags: YoutubePlayerFlags(
      autoPlay: true,
    ),
  );
  super.initState();
}

@override
Widget build(BuildContext context) {
  return YoutubePlayerBuilder(
    player: YoutubePlayer(
      controller: _controller,
      showVideoProgressIndicator: true,
      progressIndicatorColor: Colors.red,
      progressColors: ProgressBarColors(
        playedColor: Colors.red,
        handleColor: Colors.red.shade400,
        backgroundColor: Colors.grey,
      ),
      onReady: () {},
    ),
    builder: (_, player) {
      return Scaffold(
        backgroundColor: Colors.white,
        appBar: AppBarBack(),
        body: Column(children: [
          player,
          Expanded(
            child: SingleChildScrollView(
              physics: BouncingScrollPhysics(),
              child: Padding(
                padding: const EdgeInsets.symmetric(
                  horizontal: 30.0, vertical: 10.0),
                child: Column(
                  children: [
                    Text(
                      videoTitle,
                      style: LibraryStyle.headingVideo,
                    ),
                    SizedBox(
                      height: 20.0,
                    ),
                    Text(videoDescription,
                      style: LibraryStyle.videoDescription)
                  ],
                ),
              ),
            ),
          ],
        ),
      );
    },
  );
}

```

```

    ),
  ),
),
),
]),
);
},
);
}
}
}

```

Листинг one_news_screen.dart

```

import 'package:flutter/material.dart';
import 'package:it_way_app/components/appBarBack.dart';
import 'package:it_way_app/data/news_data.dart';
import 'package:it_way_app/statics/style.dart';

class OneNewsScreen extends StatefulWidget {
  const OneNewsScreen({
    @required this.data,
  });
  final NewsData data;
  @override
  _OneNewsScreenState createState() => _OneNewsScreenState();
}

class _OneNewsScreenState extends State<OneNewsScreen> {
  String get newsTitle => widget.data.newsTitle;
  String get newsBody => widget.data.newsBody;
  String get newsImage => widget.data.newsImageURL;
  String get publishedAt => widget.data.publishedAt;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.white,
      appBar: AppBarBack(),
      body: Padding(
        padding: const EdgeInsets.all(30.0),
        child: SingleChildScrollView(
          physics: BouncingScrollPhysics(),
          child: Column(
            children: [
              Image.network(newsImage),
              SizedBox(
                height: 20.0,
              ),
              Text(
                newsTitle,

```

```

        style: LibraryStyle.oneNewsHeading,
      ),
      SizedBox(
        height: 20.0,
      ),
      Text(newsBody, style: LibraryStyle.newsBody)
    ],
  ),
),
);
}
}

```

Листинг auth.dart

```

import 'package:firebase_auth/firebase_auth.dart';
import 'package:it_way_app/user.dart';

class AuthSrevice {
  final FirebaseAuth fAuth = FirebaseAuth.instance;

  Future<UserProfile> signinWithEmailAndPassword(
    String email, String password) async {
    try {
      UserCredential result = await fAuth.signInWithEmailAndPassword(
        email: email, password: password);
      User user = result.user;
      return UserProfile.fromFirebase(user);
    } catch (e) {
      print(e);
      return null;
    }
  }

  Future<UserProfile> registrWithEmailAndPassword(
    String email, String password) async {
    try {
      UserCredential result = await fAuth.createUserWithEmailAndPassword(
        email: email, password: password);
      User user = result.user;
      return UserProfile.fromFirebase(user);
    } catch (e) {
      print(e);
      return null;
    }
  }

  Future logout() async {
    await fAuth.signOut();
  }
}

```

```

}

Stream<UserProfile> get currentUser {
  return fAuth.authStateChanges().map(
    (User user) => user != null ? UserProfile.fromFirebase(user) : null);
}
}

```

Листинг profile_screen.dart

```

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:it_way_app/components/login_button.dart';
import 'package:it_way_app/components/reusable_card.dart';
import 'package:it_way_app/components/login_widget.dart';
import 'package:it_way_app/services/auth.dart';
import 'package:it_way_app/statics/navigation.dart';
import 'package:it_way_app/statics/style.dart';
import 'package:it_way_app/statics/colors.dart';
import 'package:it_way_app/statics/assets.dart';

import 'package:it_way_app/components/appBarBack.dart';
import 'package:it_way_app/user.dart';

import '../screen_navigation.dart';

class ProfileScreen extends StatefulWidget {
  @override
  _ProfileScreenState createState() => _ProfileScreenState();
}

class _ProfileScreenState extends State<ProfileScreen> {
  TextEditingController emailController = TextEditingController();
  TextEditingController passwordController = TextEditingController();
  TextEditingController userNameController = TextEditingController();

  String email;
  String password;
  String userName;
  bool showLogin = true;

  AuthService authService = AuthService();

  @override
  Widget build(BuildContext context) {
    void loginUser() async {
      email = emailController.text;
      password = passwordController.text;

```

```

    if (email.isEmpty || password.isEmpty) return;

    UserProfile user = await authService.signInWithEmailAndPassword(
        email.trim(), password.trim());
    if (user == null) {
        Fluttertoast.showToast(
            msg: 'Не удалось войти!',
            toastLength: Toast.LENGTH_SHORT,
            gravity: ToastGravity.CENTER,
            backgroundColor: Colors.red,
            textColor: Colors.black,
            fontSize: 16.0);
    } else {
        ScreenNavigation.routeTo(
            route: LibraryRoutes.userScreenRemove, context: context);
        emailController.clear();
        passwordController.clear();
    }
}

void registerUser() async {
    email = emailController.text;
    password = passwordController.text;
    userName = userNameController.text;

    if (email.isEmpty || password.isEmpty || userName.isEmpty) return;

    UserProfile user = await authService.registrWithEmailAndPassword(
        email.trim(), password.trim());
    if (user == null) {
        Fluttertoast.showToast(
            msg: 'Не удалось зарегистрироваться!',
            toastLength: Toast.LENGTH_SHORT,
            gravity: ToastGravity.CENTER,
            backgroundColor: Colors.red,
            textColor: Colors.black,
            fontSize: 16.0);
    } else {
        var ref = FirebaseFirestore.instance.doc('user/${user.id}');
        ref.set({
            'name': userName,
        });
        ScreenNavigation.routeTo(
            route: LibraryRoutes.userScreenRemove,
            context: context,
        );
        emailController.clear();
        passwordController.clear();
    }
}
}

```



```

return Scaffold(
  resizeToAvoidBottomInset: false,
  backgroundColor: Colors.white,
  appBar: AppBarBack(),
  body: Padding(
    padding: const EdgeInsets.all(30.0),
    child: Column(
      children: [
        Padding(
          padding: const EdgeInsets.only(bottom: 20.0),
          child: Column(
            children: [
              Padding(
                padding: const EdgeInsets.all(20.0),
                child: Text(
                  'Авторизация',
                  style: LibraryStyle.logInHeading,
                ),
              ),
              Text(
                'Чтобы у вас появилась возможность комментировать, необходимо авторизоваться',
                style: LibraryStyle.settings,
                textAlign: TextAlign.center,
              ),
            ],
          ),
        ),
        Column(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          children: [
            LoginWidget(
              icon: Icon(
                Icons.email,
              ),
              nameOfCard: 'Email',
              controller: emailController,
              obscure: false,
            ),
            LoginWidget(
              icon: Icon(
                Icons.lock,
              ),
              nameOfCard: 'Password',
              controller: passwordController,
              obscure: true,
            ),
            (showLogin
              ? Column(
                  children: [

```

```

LoginWidget(
  icon: Icon(
    Icons.account_circle,
  ),
  nameOfCard: 'Введите никнейм',
  controller: userNameController,
  obscure: false,
),
LoginButton(
  label: 'Зарегистрироваться',
  func: registerUser,
),
Padding(
  padding: EdgeInsets.only(top: 10.0),
  child: GestureDetector(
    child: Column(
      children: [
        Text('Уже зарегистрированы?'),
        Text(
          'Войдите.',
          style: TextStyle(
            color: LibraryColors.kActiveColor),
        ),
      ],
    ),
    onTap: () {
      setState(() {
        showLogin = false;
      });
    },
  ),
),
],
)
: Column(
  children: [
    LoginButton(
      label: 'Войти',
      func: loginUser,
    ),
    Padding(
      padding: EdgeInsets.only(top: 10.0),
      child: GestureDetector(
        child: Column(
          children: [
            Text('Еще не зарегистрированы?'),
            Text(
              'Регистрируйтесь.',
              style: TextStyle(
                color: LibraryColors.kActiveColor),
            ),
          ],
        ),
      ),
    ),
  ],
)

```



```

import 'package:it_way_app/statics/navigation.dart';
import 'package:it_way_app/statics/style.dart';
import 'package:it_way_app/statics/colors.dart';
import 'package:it_way_app/statics/assets.dart';

import 'package:it_way_app/components/appBarBack.dart';
import 'package:it_way_app/user.dart';

class UserScreen extends StatefulWidget {
  @override
  _UserScreenState createState() => _UserScreenState();
}

class _UserScreenState extends State<UserScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: false,
      backgroundColor: Colors.white,
      appBar: AppBarBack(),
      body: Padding(
        padding: const EdgeInsets.all(30.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.spaceBetween,
          children: [
            Column(
              children: [
                Padding(
                  padding: const EdgeInsets.all(20.0),
                  child: Text(
                    'Настройка профиля',
                    style: LibraryStyle.logInHeading,
                  ),
                ),
                CircleAvatar(
                  radius: 50.0,
                  backgroundColor: Colors.white,
                  backgroundImage:
                    AssetImage('assets/images/account_circle.png'),
                ),
                Align(
                  alignment: Alignment.centerLeft,
                  child: Text(
                    'Сообщество'.toUpperCase(),
                    style: LibraryStyle.settings,
                  ),
                ),
                SizedBox(height: 10.0),
                ReusableCadrInProfileScreen(
                  iconOnCard: LibraryAssets.syte,
                  nameOfCard: 'Сайт',
                ),
              ],
            ),
          ],
        ),
      ),
    );
  }
}

```

```

    ),
    ReusableCadrInProfileScreen(
      iconOnCard: LibraryAssets.telegram,
      nameOfCard: 'Чат в телеграмме',
    ),
  ],
),
GestureDetector(
  onTap: () {
    AuthSrevice().logout();
    ScreenNavigation.routeTo(
      route: LibraryRoutes.registerScreenRemove,
      context: context);
  },
  child: Container(
    height: 50.0,
    width: MediaQuery.of(context).size.width,
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(10.0),
      color: LibraryColors.kActiveColor),
    child: Center(
      child: Text(
        'Выйти',
        style: LibraryStyle.loginButtonText,
        textAlign: TextAlign.center,
      ),
    ),
  ),
),
),
],
),
),
);
}
}

```

Листинг screen_navigation.dart

```
import 'package:flutter/widgets.dart';
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';
import 'package:it_way_app/sreens/one_video_screen.dart';
import 'package:it_way_app/sreens/user_screen.dart';
import 'package:it_way_app/statics/navigation.dart';
import 'package:it_way_app/sreens/profile_screen.dart';
import 'package:it_way_app/sreens/one_podcast_screen.dart';
import 'package:it_way_app/sreens/one_news_screen.dart';

class ScreenNavigation {
  static void routeTo({
```

```

@required String route,
@required BuildContext context,
dynamic bundle,
}) {
  switch (route) {
    case LibraryRoutes.profileScreen:
      Navigator.of(context).push(
        MaterialPageRoute<Widget>(
          builder: (BuildContext context) => ProfileScreen(),
        ),
      );
      break;
    case LibraryRoutes.userScreen:
      Navigator.of(context).push(
        MaterialPageRoute<Widget>(
          builder: (BuildContext context) => UserScreen(),
        ),
      );
      break;
    case LibraryRoutes.userScreenRemove:
      Navigator.of(context).pushReplacement(
        MaterialPageRoute<Widget>(builder: (context) => UserScreen()),
      );
      break;
    case LibraryRoutes.registerScreenRemove:
      Navigator.of(context).pushReplacement(
        MaterialPageRoute<Widget>(builder: (context) => ProfileScreen()),
      );
      break;
    case LibraryRoutes.onePodcastScreen:
      Navigator.of(context).push(
        MaterialPageRoute<Widget>(
          builder: (BuildContext context) => OnePodcastScreen(
            data: bundle['data'],
          ),
        ),
      );
      break;
    case LibraryRoutes.oneVideoScreen:
      Navigator.of(context).push(
        MaterialPageRoute<Widget>(
          builder: (BuildContext context) => OneVideoScreen(
            data: bundle['data'],
          ),
        ),
      );
      break;
    case LibraryRoutes.oneNewsScreen:
      Navigator.of(context).push(
        MaterialPageRoute<Widget>(

```

```

        builder: (BuildContext context) => OneNewsScreen(
          data: bundle['data'],
        ),
      ),
    );
    break;
  case LibraryRoutes.pop:
    Navigator.of(context).pop();
    break;
  default:
  }
}
}

```

Листинг assets.dart

```

import 'package:flutter/widgets.dart';

class LibraryAssets {
  LibraryAssets._();
  static const _kFontFam = 'ITWayAppIcon';

  static const IconData syte = IconData(0xe800, fontFamily: _kFontFam);
  static const IconData telegram = IconData(0xe801, fontFamily: _kFontFam);
  static const IconData thumb_down_alt =
    IconData(0xe802, fontFamily: _kFontFam);
  static const IconData thumb_up_alt = IconData(0xe803, fontFamily: _kFontFam);
  static const IconData videoicon = IconData(0xe804, fontFamily: _kFontFam);
  static const IconData vk = IconData(0xe805, fontFamily: _kFontFam);
  static const IconData account_box = IconData(0xe806, fontFamily: _kFontFam);
  static const IconData back = IconData(0xe807, fontFamily: _kFontFam);
  static const IconData bookmark = IconData(0xe808, fontFamily: _kFontFam);
  static const IconData bookmarkicon = IconData(0xe809, fontFamily: _kFontFam);
  static const IconData contenticon = IconData(0xe80a, fontFamily: _kFontFam);
  static const IconData google = IconData(0xe80b, fontFamily: _kFontFam);
  static const IconData newsicon = IconData(0xe80c, fontFamily: _kFontFam);
  static const IconData notifications = IconData(0xe80d, fontFamily: _kFontFam);
  static const IconData send = IconData(0xe80e, fontFamily: _kFontFam);
  static const IconData share = IconData(0xe80f, fontFamily: _kFontFam);
}

```

Листинг style.dart

```

import 'package:flutter/material.dart';
import 'colors.dart';

class LibraryStyle {
  static const optionStyle =
    TextStyle(fontSize: 30, fontWeight: FontWeight.bold);
}

```

```

static const headingStyle = TextStyle(
  fontSize: 32.0, fontWeight: FontWeight.bold, color: Colors.black);

static const headingPodcastAndNews =
  TextStyle(fontSize: 18, color: Colors.white);

static const headingVideo = TextStyle(fontSize: 18, color: Colors.black);

static const subtitleStyle = TextStyle(color: LibraryColors.kUnActiveColor);

static const typeContentStyle =
  TextStyle(color: LibraryColors.kUnActiveColor, fontSize: 9.0);

static const reusableCardStyle =
  TextStyle(color: Colors.black, fontSize: 16.0);

static const logInHeading =
  TextStyle(fontSize: 24.0, fontWeight: FontWeight.bold);

static const settings =
  TextStyle(fontSize: 16.0, fontWeight: FontWeight.w500);

static const onePodcastHeading =
  TextStyle(fontSize: 18.0, fontWeight: FontWeight.w500);

static const videoDescription = TextStyle(
  fontSize: 16.0,
);
static const newsBody = TextStyle(
  fontSize: 16.0,
);

static const oneNewsHeading =
  TextStyle(fontSize: 18.0, fontWeight: FontWeight.w500);

static const loginText = TextStyle(
  fontSize: 18.0,
  color: LibraryColors.kActiveColor,
);

static const loginButtonText = TextStyle(
  fontSize: 18.0,
  color: Colors.white,
);
}

```

Листинг loading.dart

```

import 'package:flutter/material.dart';
import 'package:it_way_app/statics/colors.dart';

```



```

class LoadingApp extends StatefulWidget {
  @override
  _LoadingAppState createState() => _LoadingAppState();
}

class _LoadingAppState extends State<LoadingApp> {
  @override
  Widget build(BuildContext context) {
    return Container(
      alignment: Alignment.center,
      height: MediaQuery.of(context).size.height,
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Text(
            '«IT WAY»',
            style: TextStyle(color: LibraryColors.kActiveColor, fontSize: 24.0),
          ),
          new CircularProgressIndicator(
            color: LibraryColors.kActiveColor,
          ),
        ],
      ),
    );
  }
}

```